OPTIMAL CONTINUOUS THRUST ORBITAL EVASIVE MANEUVERS FROM GEOSYNCHRONOUS ORBIT

THESIS

Robert D. Preissinger
Captain, USAF

AFIT/GA/AA/86D-12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/AA/86D-12

OPTIMAL CONTINUOUS THRUST ORBITAL EVASIVE MANEUVERS FROM GEOSYNCHRONOUS ORBIT

THESIS

Robert D. Preissinger
Captain, USAF

AFIT/GA/AA/86D-12

Approved for public release; distribution unlimited

AFIT/GA/AA/86D-12

OPTIMAL CONTINUOUS THRUST ORBITAL EVASIVE MANEUVERS FROM GEOSYNCHRONOUS ORBIT

THESIS

Presented to the Faculty of the School of Engineering

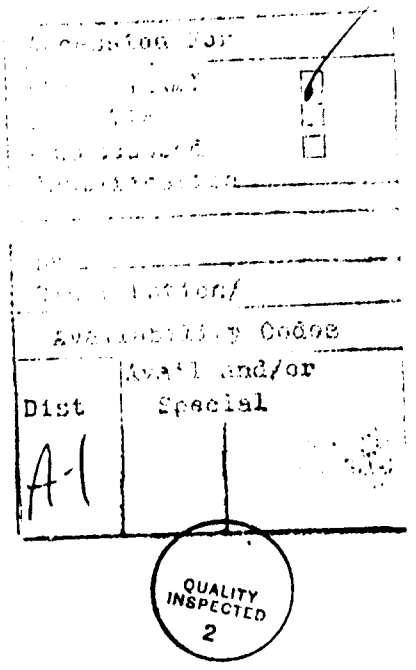
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Astronautical Engineering



Distribution Statement A is correct for this report.
Per Lt. Col. Joseph W. Widham, AFIT/ENY

Robert D. Preissinger

Captain USAF

December 1986

Approved for public release; distribution unlimited

Preface

I began this research with the goal in mind of creating a computer program that could be used tactically to find the best evasive maneuver for a satellite in geosynchronous orbit under a given attack through the use of continuous thrust. I wanted this computer program to be able to track the distance between the attacking satellite and the satellite being attacked throughout the maneuver. My intention was to use a very low thrust such as that available for use in satellite attitude control and station keeping. I was able to make substantial progress through the use of an IMSL routine known as DVCPR.

The biggest problem was in de-sensitizing the subroutine, DVCPR. The extreme sensitivity of this subroutine to co-state inputs proved to be very time consuming. Hopefully, my study will save future students some of this painstaking drudgery.

I owe a great deal of thanks to my faculty advisor, Lt Col Joseph W. Widhalm, for his expertise, confidence, and patience given to me throughout this project. However, my greatest thanks goes to my wife, Carolyn, for all of her invaluable encouragement and understanding.

Robert D. Preissinger

LOGITEC M-1601A-PC (IBM Clone); Fancyword; Star-Micronics SG-10

Table of Contents

List of Figures

List of Symbols

$\mathcal{H}$	Hamiltonian
$\mathcal{H}_u$	Gradient of the Hamiltonian --- $\partial H/\partial u$
J	Cost Function
m	mass
m'	constant mass flow rate
μ	Geocentric gravitational constant
g_o	universal gravitational constant
Th	Thrust magnitude
TPBVP	Two point boundary value problem
DU	Geocentric distance unit
TU	Geocentric time unit
t	Time
t_f	Final time
u	Steering angle measured from x axis (scalar control variable)
$\underline{\rho}$	vector measured from geocenter to satellite (for the two satellite problem, ρ is the vector between the two satellites)
ρ	magnitude of ρ vector
ρ_x	x component of ρ vector
ρ_y	y component of ρ vector
$\underline{r}_1$	vector (from geocenter) to satellite 1 (attackee)
r_1	magnitude of $\underline{r}_1$ vector
$\underline{r}_2$	vector (from geocenter) to satellite 2 (attacker)
r_2	magnitude of $\underline{r}_2$ vector

r_{1x}	x component of $\underline{r}_1$ vector
r_{2x}	x component of $\underline{r}_2$ vector
r_{1y}	y component of $\underline{r}_1$ vector
r_{2y}	y component of $\underline{r}_2$ vector
$\underline{v}$	velocity vector -- $d\rho/dt$
$\underline{v}_2$	velocity vector of satellite 2 -- dr_2/dt
$\underline{v}_1$	velocity vector of satellite 1 -- dr_1/dt
$\underline{X}$	vector of state variables
x'	first derivative of state vector (w.r.t. time)
$\underline{\lambda}$	vector of co-state variables
Φ	Optimizing equation true at the final time
ν	Lagrange multiplier
ψ	constraints of the problem
τ	time parameterization variable

Abstract

A satellite under attack by another orbiting body relying on an explosive device to effect a kill has the problem of avoiding a volume of space in which its destruction is highly likely. To avoid this volume, the attackee could use continuous low thrust, such as that provided by the electric propulsion in attitude control thrusters, if its warning time and orbital parameters were appropriate. A model is developed using optimal control theory and is solved numerically for the thrust direction using various magnitudes of thrust. The model progresses from a one satellite solution (where the threat is a fixed sphere in space) to a two satellite solution in which the distance between the threat and the target satellite is constantly updated (to avoid this same threat sphere). The results are given for several values of thrust, several time-of-flights (time of thrusting for the maneuvered satellite), the optimized time of flight, and for circular and non-circular end orbits.

OPTIMAL CONTINUOUS THRUST ORBITAL EVASIVE MANEUVERS FROM GEOSYNCHRONOUS ORBIT

I. Introduction

Much research has been done on the astrodynamics of orbital transfer and rendezvous, but very little of the current literature addresses the problem of avoiding an interception in space. With more and more military capabilities being placed into satellites (see Brandt (6)), the means of intercepting them are improving. It stands to reason that the problem of avoiding or defeating an interception also needs attention. Evasive maneuvering is one technique of protecting space assets.

Most of the current literature on orbital evasive maneuvers is concerned with impulsive thrust (8, 9, 10, 17, 19, 21, 26). The purpose of this thesis is to examine the optimal orbital maneuver that would be required by a satellite orbiting at geosynchronous altitude using a constant magnitude *continuous low thrust* propulsion system.

Even the simplest orbital maneuver may use up the satellite's limited supply of propellant, perhaps shortening its useful life or leaving it vulnerable to a second attack. It becomes obvious then, that an optimal solution is to minimize the time of propellant burn. This problem becomes synonymous with a minimum time optimization problem.

The problem is to determine the optimal thrust direction to transfer a vehicle from some point on an initial orbit, to some desired terminal orbit in minimum time, using a constant magnitude low thrust propulsion system.

This problem is approached using optimal control theory (See Bryson (7)). The thrust direction becomes the control variable and, through the equations of motion, a Two Point Boundary Value Problem (TPBVP) is formulated. For the desired result, the problem is first developed as a problem in which time is fixed and the problem is limited to two dimensions. An algorithm is developed to move a satellite from geosynchronous altitude to a maximum radius circular orbit in a specific time. There are two constraints used on the problem at this point; first a tangency constraint which requires the maneuvering satellite to be tangent to a threat sphere at t_f, and second a constraint which requires the final orbit to be circular. Then, since some fuel in this burn is being used to circularize the final orbit, the final circular orbit constraint is removed from the algorithm. This circular orbit at t_f constraint is omitted from all remaining runs. This results in a greater radius change in the same amount of time. The final part of the algorithm incorporates two satellites being monitored at the same time; one of these being the attacker, and the other being our maneuvered satellite (the target or attackee) The problem is then expanded to optimize time; thus accomplishing the intended task of this problem.

The coordinate system selected for this study differs from those selected by Smith (23) or Starr (25). Where they chose a polar coordinate system the coordinate system chosen for this thesis is an Earth–Centered Inertial Cartesian System similar to Bowman (4). The main reason for this selection is that an X–Y system is much easier to visualize than a polar system and a system in which it would be easy to "see" what is going on with the thrust direction was desired for this project. For ease in computing, the initial satellite position is chosen to lie on the X axis, and the problem is kept two dimensional. Simply stated, the initial problem as

ONE SATELLITE
COORDINATE SYSTEM

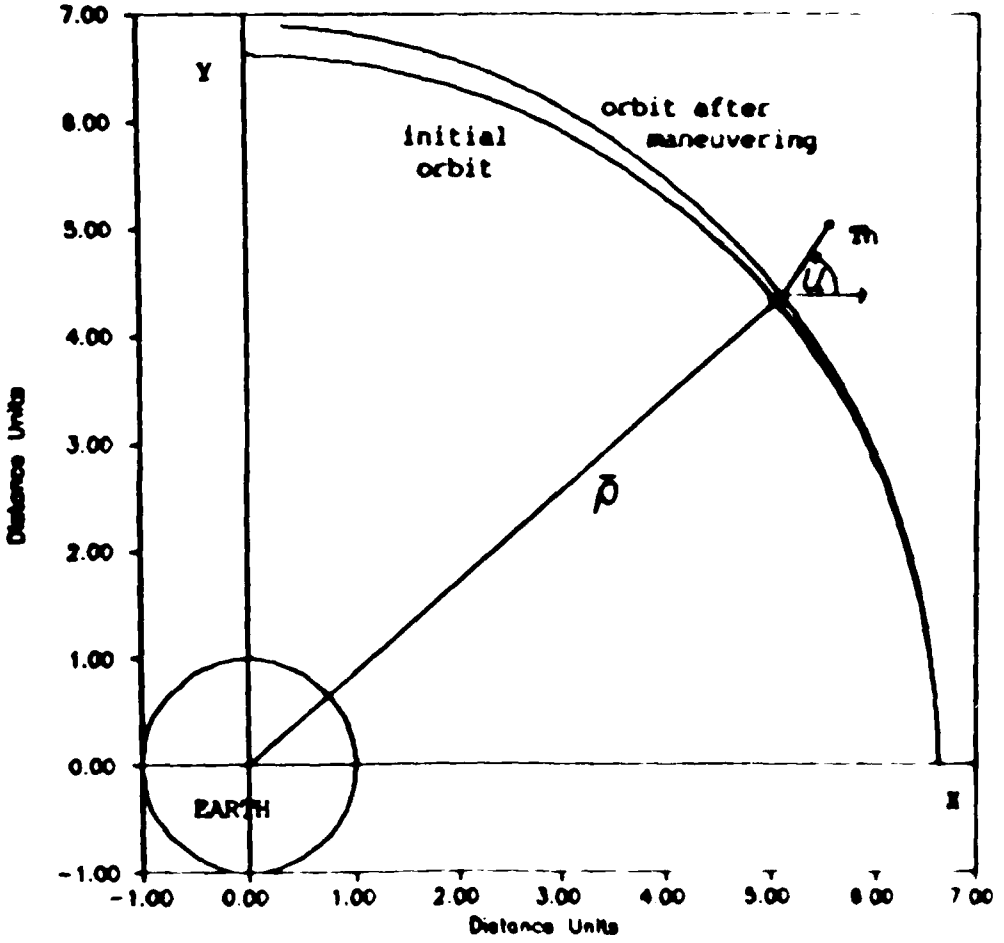


Figure 1

stated in Bryson (7:66–68) is: given a constant–thrust engine, Th = thrust, operating for a given length of time, t_f, we wish to find the thrust–direction history, u(t), to transfer a satellite from geosynchronous orbit to the largest possible circular orbit. (See figure 1)

This problem is accomplished with good results for a quarter orbit time–of–flight, and for a four hour time–of–flight. Problems encountered will be discussed in the following sections.

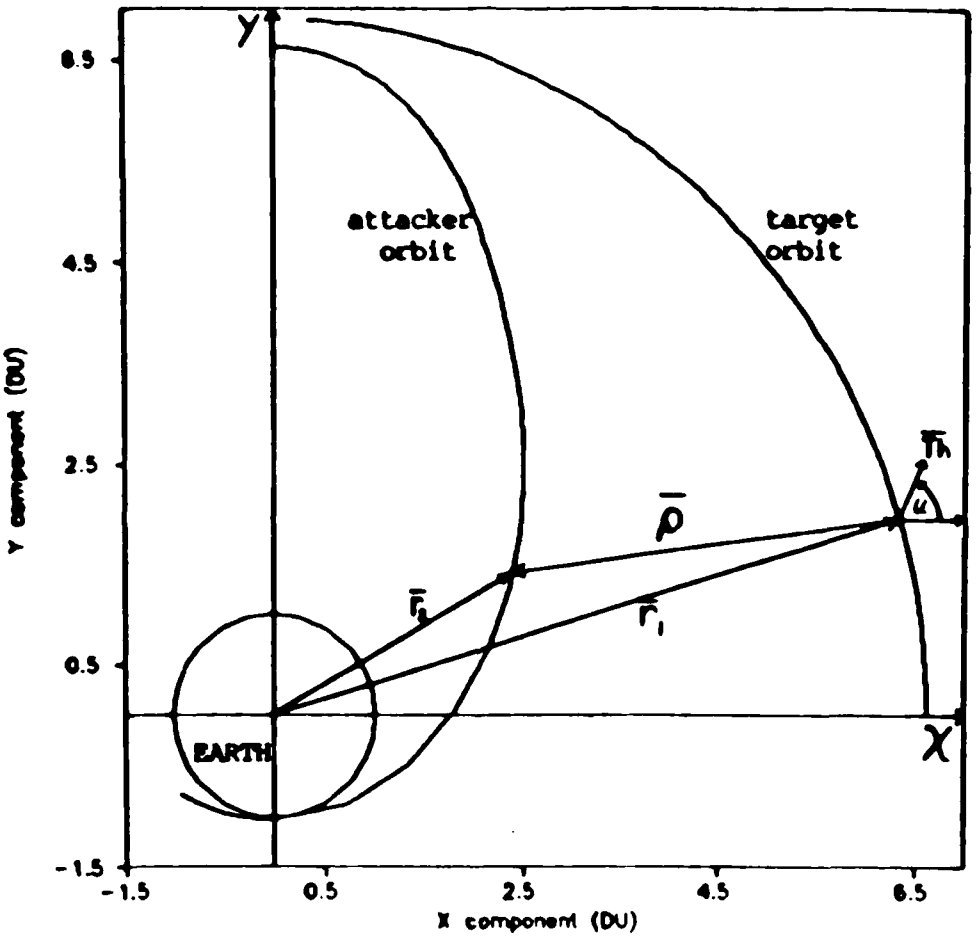
The problem is then expanded by removing the constraint of going to the largest possible circular orbit, i.e. the orbit is now just being changed to the largest possible orbit away from the threat, not to a circular orbit; however, the tangency constraint remains. This is where this problem ceases to be similar to Smith (23) or Starr (25). Results are obtained using the same thrust magnitudes and thrust durations (time–of–flight) as above.

Now comes the real objective; that of actually tracking the threat during the entire burn time, with respect to the maneuvered satellite, and optimizing the time–of–flight for the maneuver. This means beginning again with a new set of equations (derived from the initial equations of motion) for two satellites within the same problem. Time minimization is accomplished by parameterizing time in the equations of motion and applying optimization techniques to come up with the minimum time–of–flight for the total maneuver to increase satellite orbit by a specific distance. Results are also obtained for the same thrusts magnitudes and thrust durations as before.

This thesis contains all the mathematical analysis in Section II, a description of the specific computer implementation of the algorithm and the problems encountered in Section III, a discussion of the results in Section IV, and conclusion and recommendations in Section V.

In Section II, the problem equations of motion are listed for a fixed time, maximum radius increase algorithm. As stated above, one of the initial constraints is then removed and the problem is reworked to get an even greater radius increase. Therefore, it becomes possible to get a greater threat avoidance distance with this second method. The problem is then reformulated and expanded to a two satellite problem in which the distance between the two satellites can be monitored throughout the maneuver. (see figure 2) The problem of keeping track of the threat satellite throughout the burn time is also discussed. Section III discusses, in detail, the problems encountered in all aspects and versions of the algorithm, and how success is finally achieved for all specified thrust magnitudes and burn times. In Section IV, the results are discussed for various thrust magnitudes and thrust durations with the relationship between the two. The results of the time minimization problem are also discussed in this section. Finally, the conclusions of this thesis are discussed along with some of the tactics of orbital evasion and some suggestions for further work in this area.

TWO SATELLITE COORDINATE SYSTEM



$$\rho = r_2 - r_1$$

Figure 2.

II. PROBLEM FORMULATION

The equations of motion are developed from two-body motion utilizing the given coordinate systems. (see figs. 1 and 2) In order to keep things as simple as possible, perturbations are neglected and only two-body orbital dynamics are considered. The basic equations come from Bate (3) and Baker (1). The only forces acting on the satellite become thrust and gravity. The angle between the thrust vector and the X axis becomes the control.

Single Satellite Problem

For the single satellite problem ρ is the radius of the satellite measured from the center of the earth. In the following equations ρ_x becomes x_1, ρ_y becomes x_2, and so on. The resulting state equations are:

$$x'_1 = x_3 \tag{1}$$

$$x'_2 = x_4 \tag{2}$$

$$x'_3 = -\mu x_1/\rho^3 + Th/(m_o - m't)COS(u) \tag{3}$$

$$x'_4 = -\mu x_2/\rho^3 + Th/(m_o - m't)SIN(u) \tag{4}$$

with initial conditions of:

$$x_1(0) = 6.622791181 \text{ DU} \tag{5}$$

$$x_2(0) = 0.0 \text{ DU} \tag{6}$$

$$x_3(0) = 0.0 \text{ DU/TU} \tag{7}$$

$$x_4(0) = 0.3885792278 \text{ DU/TU} \tag{8}$$

The cost function for this problem then becomes

$$J = \rho(t_f) \tag{9}$$

The Hamiltonian of the system is, therefore

$$\mathcal{H} = \lambda_1 x_3 + \lambda_2 x_4 + \lambda_3(-\mu x_1/\rho^3 + Th/(m_o - m't)COS(u)) + \lambda_4(-\mu x_2/\rho^3 + Th/(m_o - m't)SIN(u)) \tag{10}$$

Minimizing the Hamiltonian with respect to u results in:

$$\mathcal{H}_u = -\lambda_3(Th/(m_o - m't))SIN(u) + \lambda_4(Th/(m_o - m't))COS(u) \tag{11}$$

Therefore, the resulting control is:

$$TAN(u) = \lambda_4/\lambda_3 \tag{12}$$

The resulting co-state equations will now be given with the following representations:

$$x_5 = \lambda_1 \quad x_6 = \lambda_2 \quad x_7 = \lambda_3 \quad x_8 = \lambda_4 \tag{13}$$

$$COS(u) = x_7/(1+x_8^2/x_7^2)^{1/2} \tag{14}$$

$$SIN(u) = x_8/(1+x_8^2/x_7^2)^{1/2} \tag{15}$$

giving the co-state equations as:

$$x'_5 = \mu[x_7/\rho^3 - 3x_7x_1^2/\rho^5 - 3x_8x_2x_1/\rho^5] \quad (16)$$

$$x'_6 = \mu[x_8/\rho^3 - 3x_8x_2^2/\rho^5 - 3x_7x_2x_1/\rho^5] \quad (17)$$

$$x'_7 = -x_5 \quad (18)$$

$$x'_8 = -x_6 \quad (19)$$

The remaining boundary conditions come from the two constraints on the problem at t_f and from the necessary conditions as found from the Φ equation. The first constraint forces a tangency of the maneuvering satellite to an imaginary threat sphere at the final time; whereas, the second implements a circular orbit at t_f.

$$\psi_1 = x_1x_5 + x_2x_4 = 0 \quad (20)$$

$$\psi_2 = (\mu/\rho(t_f))^{1/2} - v(t_f) = 0 \quad (21)$$

where:

$$v = (x_3^2 + x_4^2)^{1/2} \quad (22)$$

$$\Phi = |\rho| + \nu_1(\psi_1) + \nu_2(\psi_2) \quad (23)$$

(Note: $\lambda_i(t_f) = \partial\Phi/\partial x_i$ (i = 1-4)) Therefore the necessary conditions become

$$\lambda_1(t_f) = x_5(t_f) = x_1/\rho + \nu_1x_5 - \nu_2x_1\mu^{1/2}/\rho^{5/2} \quad (24)$$

$$\lambda_2(t_f) = x_6(t_f) = x_2/\rho + \nu_1x_4 - \nu_2x_2\mu^{1/2}/\rho^{5/2} \quad (25)$$

$$\lambda_3(t_f) = x_7(t_f) = \nu_1 x_1 - \nu_2 x_3/v \tag{26}$$

$$\lambda_4(t_f) = x_8(t_f) = \nu_1 x_2 - \nu_2 x_4/v \tag{27}$$

and the necessary conditions yield two more boundary conditions (at t_f):

$$x_5 x_1 + x_6 x_2 - \rho - x_7 x_3 - x_8 x_4 = 0 \tag{28}$$

$$x_5 x_1 + x_6 x_2 - \rho - \mu^{1/2}(x_7 x_3 + x_8 x_4)/(v\rho^{1/2}) = 0 \tag{29}$$

These last two boundary conditions come from algebraic manipulation of the $\lambda(t_f)$ equations when one eliminates the ν's.

Two Satellite Problem (see figure 2)

The two satellite problem equations are very similar to the single satellite problem except that they utilize the vector between the two satellites as the position state vector instead of a single satellite's position vector. This is done in order to observe what effect the dynamics of the attacking satellite has on the maneuvering problem. The following representations are used:

$$\underline{\rho} = \underline{r}_2 - \underline{r}_1, \qquad \underline{\rho}' = \underline{v}_2 - \underline{v}_1 \tag{30}$$

$$x_1 = \rho_x, \quad x_2 = \rho_y, \quad x_3 = \rho'_x, \quad x_4 = \rho'_y \tag{31}$$

The resulting state equations are:

$$x'_1 = x_3 \tag{32}$$

$$x'_2 = x_4 \tag{33}$$

$$x'_3 = -\mu r_{2x}/r_2^3 + \mu r_{1x}/r_1^3 - Th/(m_o - m't)\mathrm{COS}(u) \tag{34}$$

$$x'_4 = -\mu r_{2y}/r_2^3 + \mu r_{1y}/r_1^3 - Th/(m_o - m't)\mathrm{SIN}(u) \tag{35}$$

where r_2 was found through the use of a Kepler orbit determiner and r_1 was then found by subtracting the state variables from the appropriate components of the r_2 vector. i.e.:

$$r_1 = (r_{2x} - x_1)i + (r_{2y} - x_2)j \tag{36}$$

The following become the final problem's initial conditions:

$$x_1(0) = -7.552551 \text{ DU} \tag{37}$$

$$x_2(0) = -.7744873 \text{ DU} \tag{38}$$

$$x_3(0) = 1.0241614 \text{ DU/TU} \tag{39}$$

$$x_4(0) = -.9642712 \text{ DU/TU} \tag{40}$$

(Note: These are the result of satellite 1 being in a geosynchronous orbit and satellite 2 being in an eccentric orbit with its perigee altitude ~100 nm and its apogee altitude being geosynchronous.)

The cost function of this system remains:

$$J = \rho(t_f) \tag{9}$$

and the Hamiltonian of this system is, therefore,

$$\begin{aligned}\mathcal{H} = {} & \lambda_1 x_3 + \lambda_2 x_4 + \lambda_3(-\mu r_{2x}/r_2^3 + \mu r_{1x}/r_1^3 - Th/(m_o - m't)COS(u)) \\ & + \lambda_4(-\mu r_{2y}/r_2^3 + \mu r_{1y}/r_1^3 - Th/(m_o - m't)SIN(u))\end{aligned} \tag{41}$$

Minimising this Hamiltonian with respect to u results in the same expression as before:

$$\mathcal{H}_u = \lambda_3(Th/(m_o - m't))SIN(u) - \lambda_4(Th/(m_o - m't))COS(u) \quad (11)$$

Therefore, the control remains:

$$TAN(u) = \lambda_4/\lambda_3 \quad (12)$$

The co-state equations will remain given with the following representations:

$$x_5 = \lambda_1 \quad x_6 = \lambda_2 \quad x_7 = \lambda_3 \quad x_8 = \lambda_4 \quad (13)$$

$$COS(u) = x_7/(1+x_8^2/x_7^2)^{1/2} \quad (14)$$

$$SIN(u) = x_8/(1+x_8^2/x_7^2)^{1/2} \quad (15)$$

giving the co-state equations as:

$$x'_5 = \mu[x_7/r_1^3 - 3x_7r_{1x}^2/r_1^5 - 3x_8r_{1x}r_{1y}/r_1^5] \quad (42)$$

$$x'_6 = \mu[x_8/r_1^3 - 3x_8r_{1y}^2/r_1^5 - 3x_7r_{1x}r_{1y}/r_1^5] \quad (43)$$

$$x'_7 = -x_5 \quad (44)$$

$$x'_8 = -x_6 \quad (45)$$

The remaining boundary conditions come from the constraint on the problem at t_f and from the necessary conditions. Once again, the constraint forces a tangency to the threat sphere at t_f.

$$\psi_1 = x_1x_8 + x_2x_4 = 0 \tag{20}$$

where:

$$\Phi = \rho + \nu_1(\psi_1) \tag{46}$$

the necessary conditions become

$$\lambda_1(t_f) = x_6(t_f) = x_1/\rho + \nu_1 x_8 \tag{47}$$

$$\lambda_2(t_f) = x_6(t_f) = x_2/\rho + \nu_1 x_4 \tag{48}$$

$$\lambda_3(t_f) = x_7(t_f) = \nu_1 x_1 \tag{49}$$

$$\lambda_4(t_f) = x_8(t_f) = \nu_1 x_2 \tag{50}$$

and the necessary conditions yield the remaining three boundary conditions (at t_f):

$$x_1x_8 - x_7x_2 = 0 \tag{51}$$

$$x_8x_7 + x_4x_8 = 0 \tag{52}$$

$$x_8x_1 + x_8x_2 - \rho = 0 \tag{53}$$

Once again, these last three boundary conditions come from algebraic manipulation of the $\lambda(t_f)$ equations when one eliminates ν_1.

III. PROBLEM SOLUTION

To add validity to the results of this thesis, this section begins with a description of how the algorithm actually works and how it was verified. Then the specific problem areas encountered in the running of all the different algorithm versions using the different constraints are discussed.

Algorithm and Verification

The algorithm is written (all versions) in FORTRAN 77. It utilizes the IMSL routine DVCPR to solve the problems in question. It was run on the VAX 11/785 Scientific Support Computer (SSC) under the Unix operating system at the School of Engineering of the Air Force Institute of Technology. The algorithm requires the first derivatives of the equations of motion as well as the first derivatives of the co-state variable equations. The first derivatives of the equations of motion are multiplied by Lagrange multipliers to form the Hamiltonian of the system. The first derivatives of the co-state variables (the Lagrange multipliers) come directly from this Hamiltonian equation as stated in Bryson (7). Since there are four equations of motion involved in the first part of the problem and four co-state variables that come from the Hamiltonian of the system, the algorithm utilizes eight equations as input to the IMSL routine. These equations are found in the subroutine FCNI. This results in an 8 × 8 Jacobian matrix ($\partial \dot{x}_i / \partial x_j$ i=1,8 j=1,8) This Jacobian matrix is found in the subroutine FCNJ. The system boundary conditions are utilized in the subroutine FCNB.

The basic discretization used is the trapezoidal rule over a possibly non−uniform mesh. This mesh is chosen adaptively, in order to make the local error of approximately the same size everywhere. Higher order discretizations are obtained by deferred corrections. Global error estimates are produced in order to control the computation. The resulting nonlinear algebraic system is solved by Newton's method, with step control, while the linearized sparse system is solved by a special form of Gaussian elimination that preserves the sparseness.

The algorithm was verified by inputting the required data for running a simple two body problem with the boundary conditions split between the initial and final time. This problem was run without thrust for six hours duration from geosynchronous altitudes and velocities and the output resulted in exactly a two body Keplerian orbit. The final stage of this thesis incorporates a Kepler orbit determiner to find the position of the attacking satellite during the entire thrust duration. After verification of this orbit determiner, (through the use of various examples from Bate (1:195−210)) it was also used to verify the two body results obtained through this algorithm.

Single Precision Version

Equations (1) through (4) and (16) through (19) were used as the XPRIME equations for the subroutine FCNI. This resulted in an 8 × 8 Jacobian matrix that was used in the subroutine FCNJ. Equations (5) through (8), (20), (21), (28), and (29) were used in the subroutine FCNB as the boundary conditions.

Several problems arose in obtaining numerical results using a single precision version of DVCPR. First, the initial definition of the control was giving divergent solutions. The computer was not giving the correct values

because the state equations needed COS(u) and SIN(u) inputs. The initial definitions were:

$$\mathrm{COS}(u) = x_7/(x_8^2 + x_7^2)^{1/2} \tag{14}$$

$$\mathrm{SIN}(u) = x_8/(x_8^2 + x_7^2)^{1/2} \tag{15}$$

While mathematically correct, this gave some problems in the computer algorithm as they caused the algorithm solution to diverge when first attempted. This was rectified in two ways. First, u was redefined as follows:

$$u = \mathrm{ATAN2}(x_8, x_7) \tag{54}$$

This utilized the intrinsic ATAN2 function of FORTRAN 77. The other way was to painstakingly take care in choosing the input grid values of the co-state variables. It is interesting to note that these two methods resulted in exactly the same results; however, the first solution was far less time consuming.

The second and biggest problem in using DVCPR for this problem was that I had to "initialize the grid" or give the Newton algorithm a good starting point. Picking good approximations for the initial state equations was not hard, but getting a good guess for the co-state values proved very time consuming, as DVCPR proved to be very sensitive to these initial guesses. (This was the most time consuming part of this entire thesis.) This problem was finally rectified by utilizing many computer runs with various co-state variable "guesses" for the initial grid variables. This was repeated until the output began "looking" close to the desired results, as

predicted from analysis of output from Smith (23) and Bowman(4). It was then possible to develop better "guesses" based on the previous output and sort of slowly "walk" toward a guess from which the algorithm would converge. Once there was a successful (i.e. converged) solution, it would be used as the initial guess for subsequent runs.

The third problem was that of convergence. DVCPR also proved to be very sensitive to the TOL (tolerance) parameter. Small values of tolerance in DVCPR caused an error message indicating that the Newton algorithm diverged, even though the solution appeared completely reasonable. Playing around with this parameter, i.e. raising TOL from 10^{-6} to 10 satisfied this convergence problem without affecting the actual output tolerances too much. No explanation can be given for this unusual behavior, as DVCPR does not document the tolerance algorithm in sufficient detail; however, this became one of the primary reasons for developing the double precision algorithms.

Another problem was a lack of convergence for any maneuver (time–of–flight) that was over approximately six and one–half hours in length. Utilization of only one quarter (1/4) of an orbit for the runs avoided this problem. Justification for this thinking is that in an actual ASAT type of defense (in the future), probable warning time will be between four and six hours (see Bowman (4), Brandt (6), Wagner (27), or Zazworsky (30)) One quarter of a geosynchronous orbit is six hours, therefore six hours was taken as the maximum amount of time to maneuver in this problem.

The final problem encountered was one of checking the results. In order to do this, results were compared with those obtained by Bowman (4). Their results utilized thrust only in the tangential direction and their coordinate system was slightly different; however, the results obtained here

compare favorably with their study. Results also compare very well with those obtained by Smith (23) and Starr (25).

Double Precision Version

It became obvious very quickly that single precision results were inconsistent and incomplete. (see figures 3–6) Therefore a double precision version of the algorithm was developed for all remaining work. This seemed to rectify most of the above mentioned problems and orbit radius was established to a 10^{-6} meter tolerance. The biggest problem remained however; "initializing the input grid" for the co–state values still proved very time consuming. Once again, guessing until the results were close and then using very small variations of the co–state variables soon produced convergent solutions. The results proved to be consistent with those obtained by Bowman (4) and Smith (23). The algorithm was initially run with the tangency constraint and the circular final orbit constraint. (See equations 20 and 21) The circular orbit constraint was then dropped to allow for a larger (greater radius) orbit in the same amount of burn time. (i.e. *more* distance with *no more* fuel than before) Results for both circular and non–circular final orbits are discussed. Boundary conditions are given in the subroutine FCNB and are included in the appendix.

Two Satellite Problem

Up until now, this thesis has dealt with increasing a single satellite's radius to the maximum attainable in a given time subject to the constraints of equations 20 and 21 and later dropping the constraint of equation 21. However, what this thesis was intended to do is to increase or maximize a distance away from a given threat satellite in an optimum time. This means that we must introduce this threat satellite into the problem, and somehow

begin determining just how far away from it our own satellite is. At any given time, as we maneuver our satellite to avoid the threat, it would be nice to observe the effects on the problem presented by the attacking satellite's orbital dynamics. This section deals with two satellites, a threat or attacking satellite and a maneuvering or target satellite.

First a Keplerian orbit determining algorithm had to be developed. This orbit determiner was derived from Bate (1:195–205) and incorporated into the main algorithm in every instance where the state variable was to be found. This orbit determiner found $\underline{r}_2$ at each time step, and $\underline{r}_1$ was then found by subtracting the appropriate state variable from the proper component of $\underline{r}_2$. (See equation 36) This allows us to treat $\underline{r}_2$ as non–time varying in the equations of motion.

Problems encountered in this version of the problem at first seemed to be insurmountable. The algorithm was initialized with previous output, in the hopes of a quick convergence. The state variable initialization was simply a matter of subtracting the previous state variable from the output of the Keplerian orbit determiner subroutine. (see equation 25) However, the co–state variable values follow no logical pattern. Once again, guess after guess finally brought a converged solution which could then be used in subsequent runs.

This problem was first run with the threat satellite in the exact same orbit as the maneuvering satellite. In other words, the threat and the target begin at the same point in space in the same orbit. The target satellite then maneuvers with its constant low thrust system, thereby changing its orbit. This orbit changes with time, and the radius is constantly increasing; or the distance from the threat is constantly increasing.

These results were then used to initialize further runs in which the threat satellite's orbit is changed to various attack runs. The final run has the threat in a high eccentricity orbit with a perigee altitude of 100 nautical miles and an apogee altitude equal to geosynchronous.

Time Minimization Version

The real problem statement was to find the optimum time to transfer evasively away from a threat. This was accomplished by parameterizing time in the original algorithm and using optimal control theory on the equations of motion. The cost function now changes from that of equation 9 to become

$$J = \rho(t_f) + \int_0^{t_f} dt \tag{55}$$

where we fix $\rho(t_f)$ to a specific magnitude.

This results in an additional state equation:

$$x'_9 = 0 \tag{56}$$

This comes about by allowing orbit transfer time to vary between zero and one, letting:

$$t = x_9 * \tau \tag{57}$$

Because $\rho(t_f)$ is fixed, the effective J becomes,

$$J_{eff} = \int_0^1 d\tau \tag{55a}$$

In order to implement this extra state equation, each of the previous state equations must now be multiplied by the new state variable (x_9) and equation (56) becomes the ninth equation needed in the subroutine FCNI.

This also brings an additional boundary condition into the problem, the Hamiltonian of this new system (now different from equation 9) is now equal to zero.

$$\mathcal{H} = [\lambda_1 x_8 + \lambda_2 x_4 + \lambda_3(-\mu x_1/\rho^3 + Th/(m_o - m't)COS(u)) + \lambda_4(-\mu x_2/\rho^3 + Th/(m_o - m't)SIN(u))] * x_9 + x_9 = 0 \qquad (58)$$

Equation 58 is included as one of the boundary conditions in the subroutine FCNB. These boundary conditions can be seen in the appendix. Since there is an additional state equation, the Jacobian is now a 9×9 matrix, and is used by the DVCPR subroutine FCNJ. The problem of grid initialization was still present, but by utilizing previous data as initial guesses for the initial grid, much time was saved in this parameterized time problem algorithm.

IV. RESULTS

This section discusses all of the separate computer runs that were accomplished for this problem. The first part discusses the single precision version of the algorithm. Double precision versions follow with both end time constraints discussed separately. Then the two satellite problem and the different attacker orbits that were utilized is discussed, and the final portion is devoted to the time parameterization part of this thesis.

Single Precision version.

The initial version of the algorithm was written in single precision. Four different runs were accomplished, $.001g_o$ thrust and $.0005g_o$ thrust, each utilizing a four hour and a six hour burn time. Each of these runs were subject to the constraints of equations 20 and 21. The boundary conditions for these runs are those found in equations 5 through 8, 20, 21, 28, and 29. The results are shown in figures 3, 4, 5 and 6 showing plots of u (Thrust direction) vs time. As stated previously, comparison of these results with those obtained by Bowman (4) is very favorable and shows that for thrust magnitudes in the range of $.0005g_o$ up to $.001g_o$, significant orbit changes are possible. This algorithm produced orbit radius changes greater than those obtained by Bowman (4) which was expected since Bowman dealt with thrust only in the tangential direction. Thrust levels less than this have little or no effect on orbit radius. (NOTE: g_o is the gravitational constant for the earth $-1DU/TU^2-$or$-32.1466m/s^2$)

THRUST DIRECTION (u) vs TIME -- .001g_o Thrust -- 6 HR TOF

Single Precision, Single Satellite

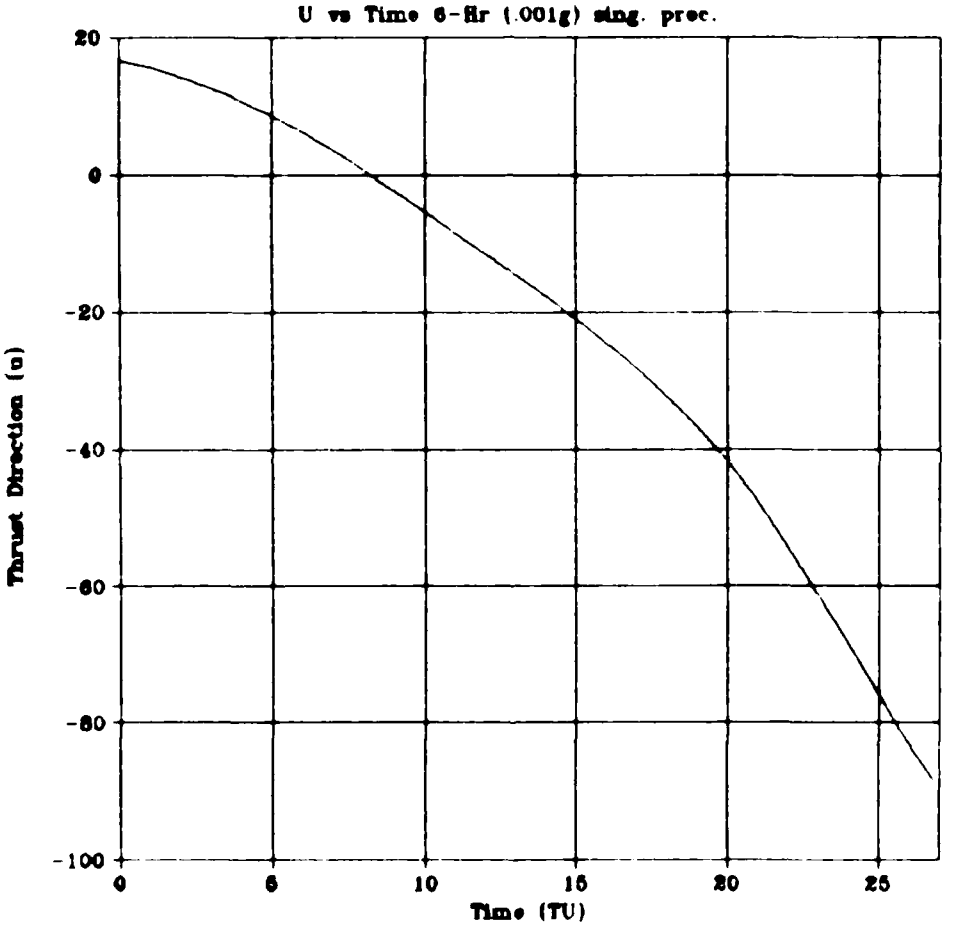


figure 8.

THRUST DIRECTION (u) vs TIME -- .0005g_o Thrust -- 6 HR TOF

Single Precision, Single Satellite

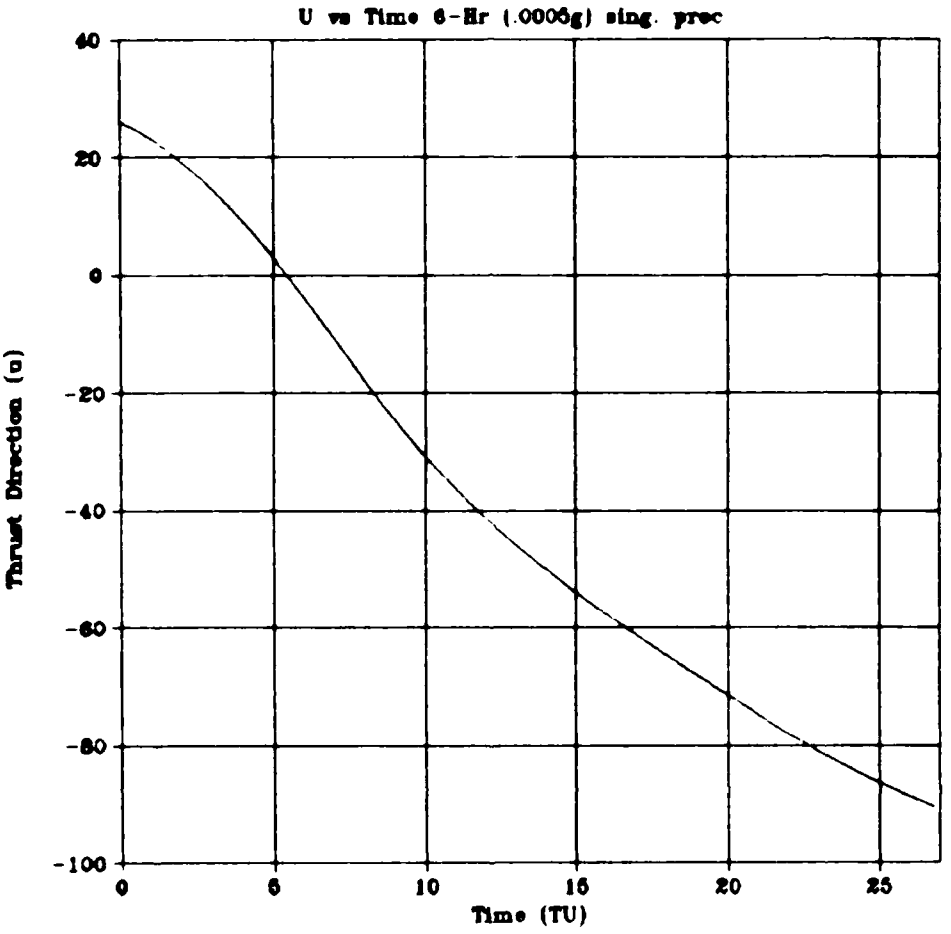


figure 4.

THRUST DIRECTION (u) vs TIME -- .001g_o Thrust -- 4 HR TOF

Single Precision, Single Satellite

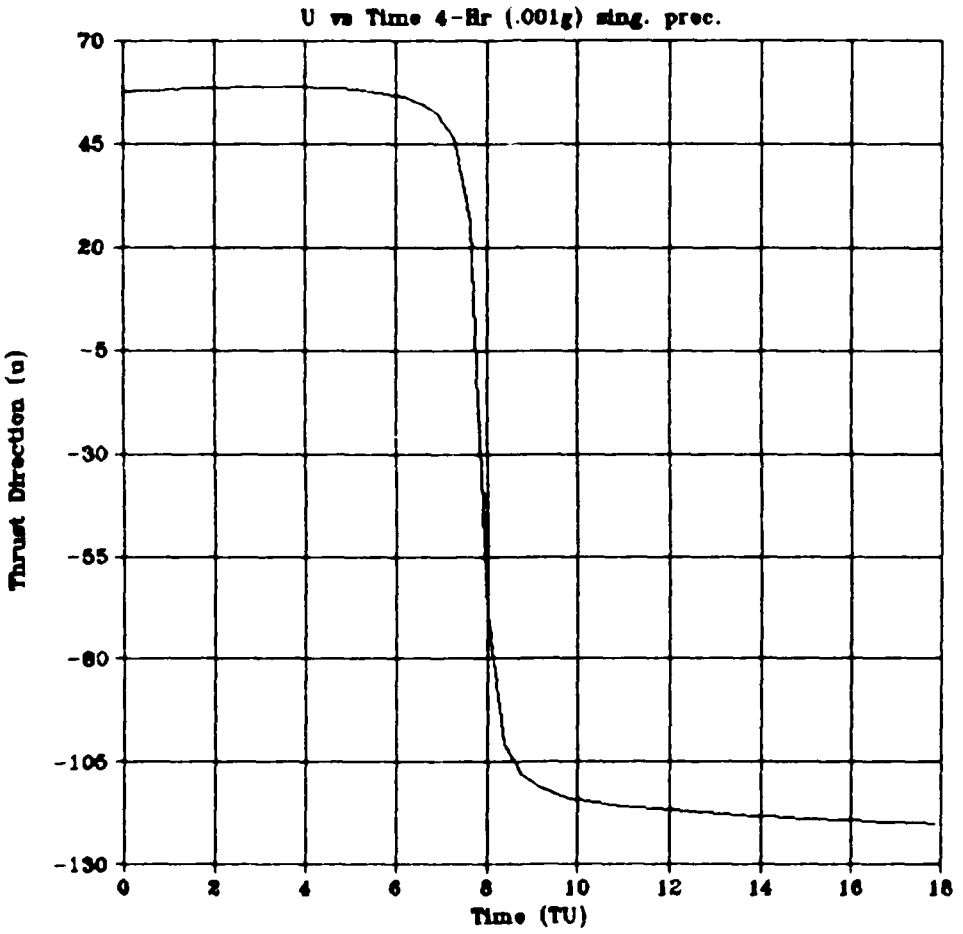


figure 5.

THRUST DIRECTION (u) vs TIME -- .0005g_o Thrust -- 4 HR TOF

Single Precision, Single Satellite

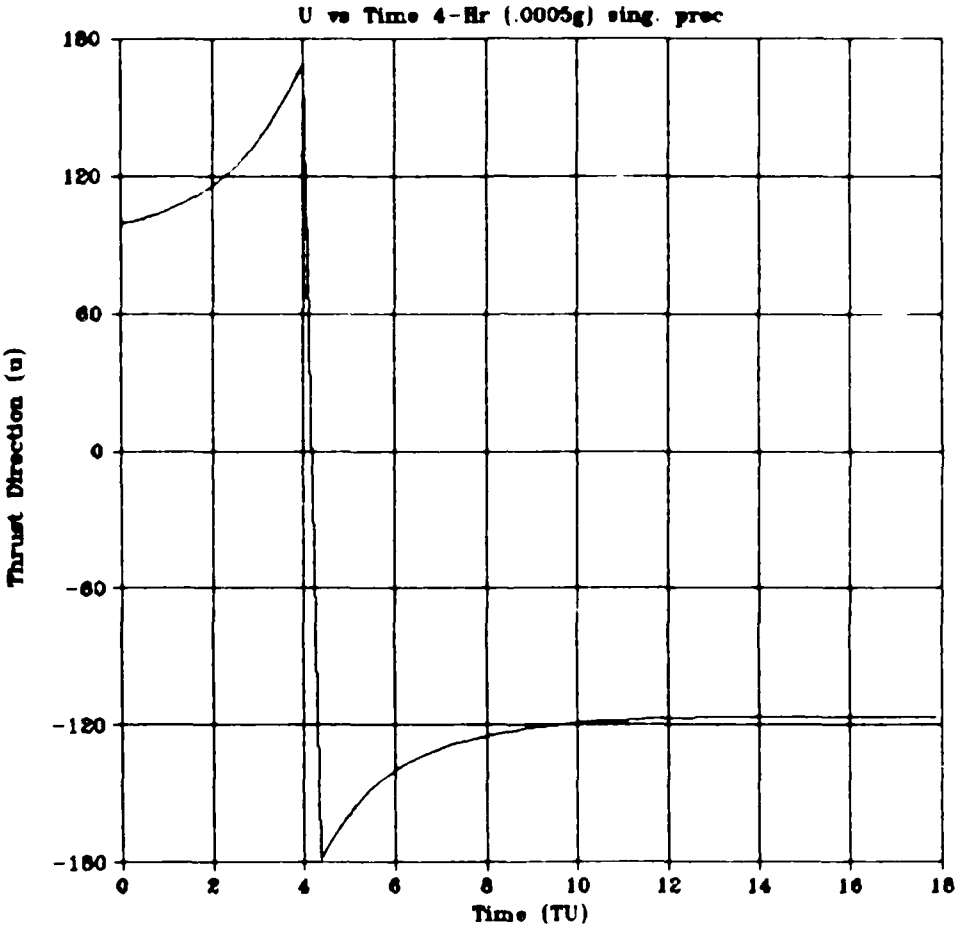


figure 6.

Orbit radius was increased by ~462 kilometers in the .0005g_o case with a 6 hour time-of-flight. (1/4 orbit period) In the .001g_o case, 6 hour time-of-flight, orbit radius increased by ~606 kilometers. This proved to be the largest amount achievable with the algorithm as written in single precision. For the .0005g_o case orbit radius increased only ~355 kilometers. This was the least amount of the algorithm. In the .001g_o, 4 hour case, the increase was ~494 kilometers.

From the graphical results, one sees a lack of consistency in the thrust direction vs the thrust time. The four hour graphs show a rapid change in thrust direction; however, the change from a positive thrust angle to a negative thrust angle seems to follow no set pattern. In the six hour graphs this transition seems to be almost linear. Although the radius increases obtained with this version seem to be valid the thrust direction angle results proved to be unacceptable. This coupled with the before mentioned problem with the TOL parameter prompted a double precision version of the algorithm to be incorporated for all additional runs and all further results are from the double precision versions.

Double Precision versions

The double precision versions of the algorithm alleviated the above mentioned problems with thrust direction angle irregularities. Two versions of the algorithm are included because the first has the constraint on the problem of the final orbit being circular (see equation 21); whereas the second version does not require a circular orbit at the end of the burn. Both versions require the maneuvering satellite to be tangent to the threat sphere at t_f (see equation 20). Boundary conditions for these runs are found in equations 5 through 8, 20, 21, 28, and 29 for the circular orbit constraint

runs and equations 5 through 8, 20, and 51 through 53 for the remaining runs. (circular constraint removed)

Circular Orbit Constraint at t_f

The first double precision version of the algorithm was run with the same constraints as the single precision version,

$$\psi_1 = x_1x_3 + x_2x_4 = 0 \tag{20}$$

$$\psi_2 = (1/\rho(t_f))^{1/2} - v(t_f) = 0 \tag{21}$$

The boundary conditions are as stated above. The problem was run with the same thrust magnitudes and durations as before. Accuracies were increased through the TOL parameter to 10^{-6} meters. (the final TOL value was 10^{-12}) The results can be seen graphically in figures 7, 8, 9, and 10.

It was interesting that for the same parameters as in the single precision version, the radius increase values were different. For the $.001g_o$ thrust, six hour burn, the radius was increased ~792 kilometers in this version (~184 km *more* than the single precision version). However for the $.0005g_o$ thrust, six hour burn, the radius increase was only ~392 kilometers in this version (~70 km *less* than the single precision version). The other two cases had similar results; the $.001g_o$ thrust, four hour burn, radius increase was ~472 kilometers (~12 km less that the single precision version), and the $.0005g_o$ thrust, four hour burn had a radius increase of only ~240 kilometers (115 km less that the sing. prec. version). While these values proved to be much more accurate than the single precision version, they were not significant changes. The good news about this version, as can be seen on the graphs, is that the thrust direction now seemed to be consistent.

THRUST DIRECTION (u) vs TIME -- .001g_o Thrust -- 6 HR TOF

Double Precision, Single Satellite, Circular end Orbit

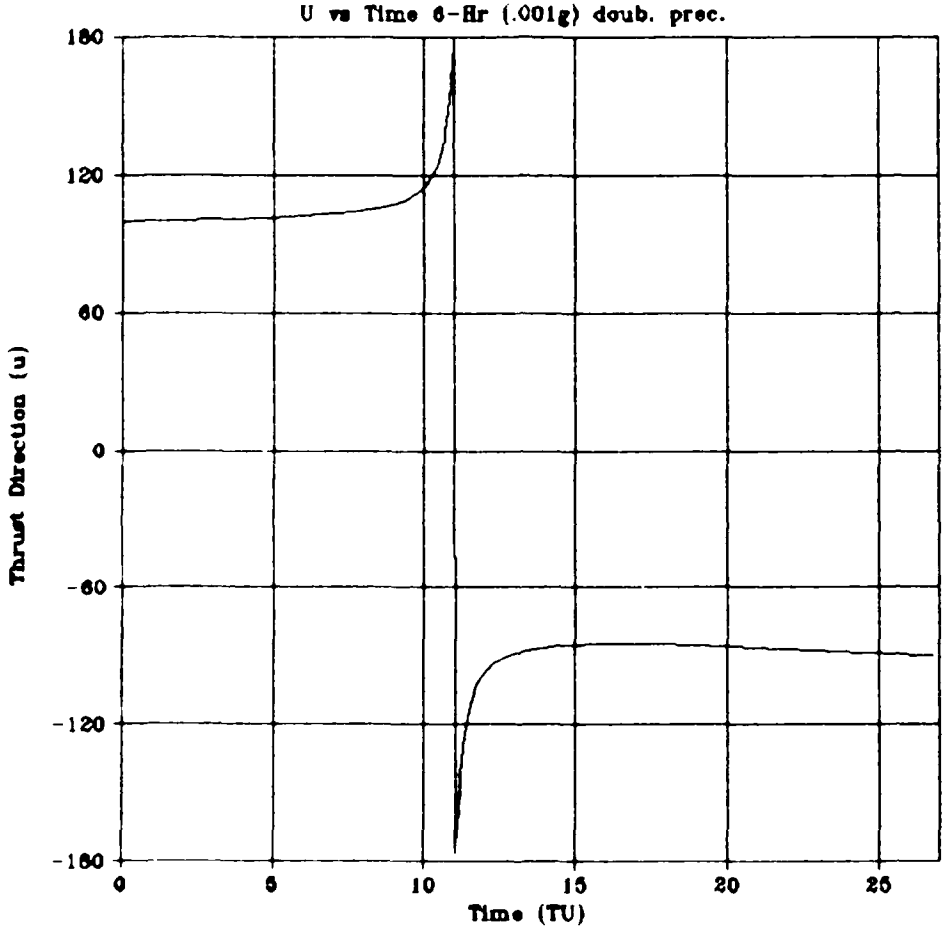


figure 7.

THRUST DIRECTION (u) vs TIME -- .0005g_o Thrust -- 6 HR TOF

Double Precision, Single Satellite, Circular end Orbit

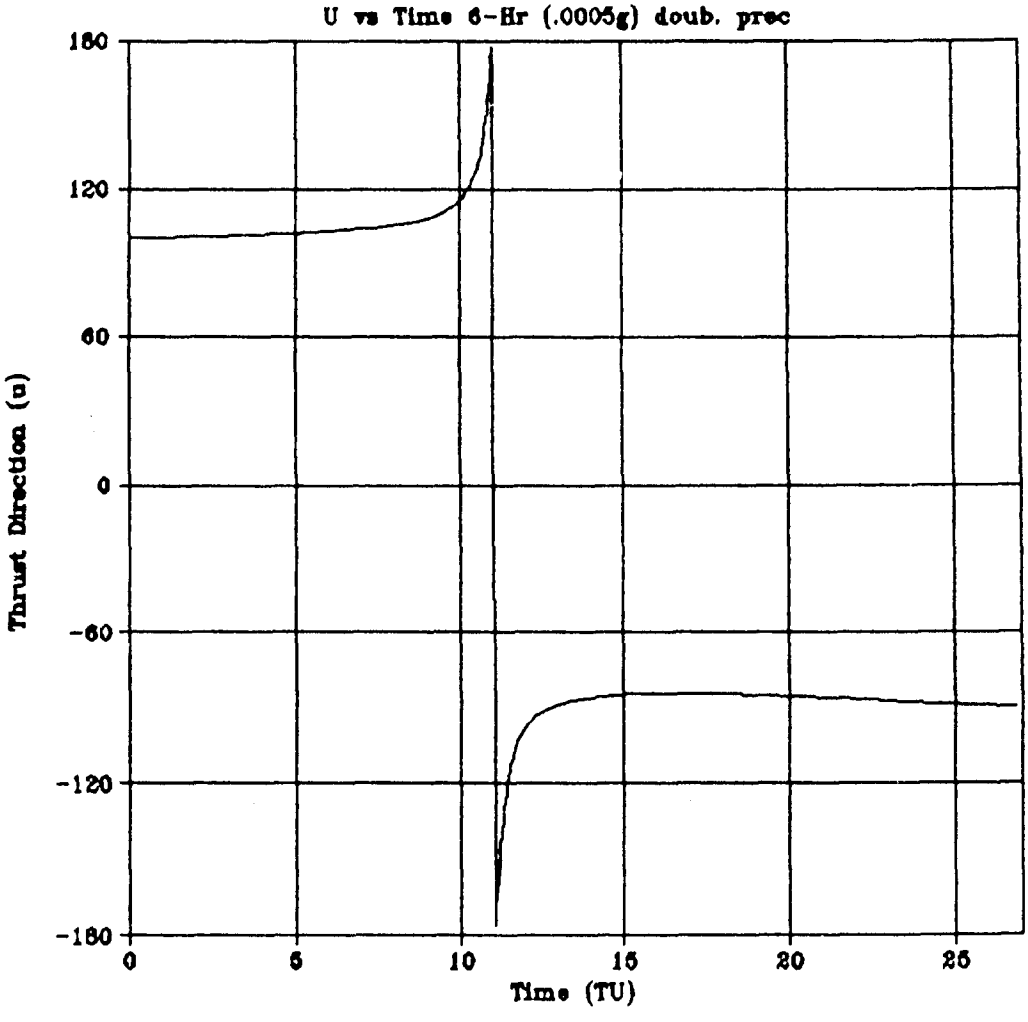


figure 8.

THRUST DIRECTION (u) vs TIME -- .001g_o Thrust -- 4 HR TOF

Double Precision, Single Satellite, Circular end Orbit

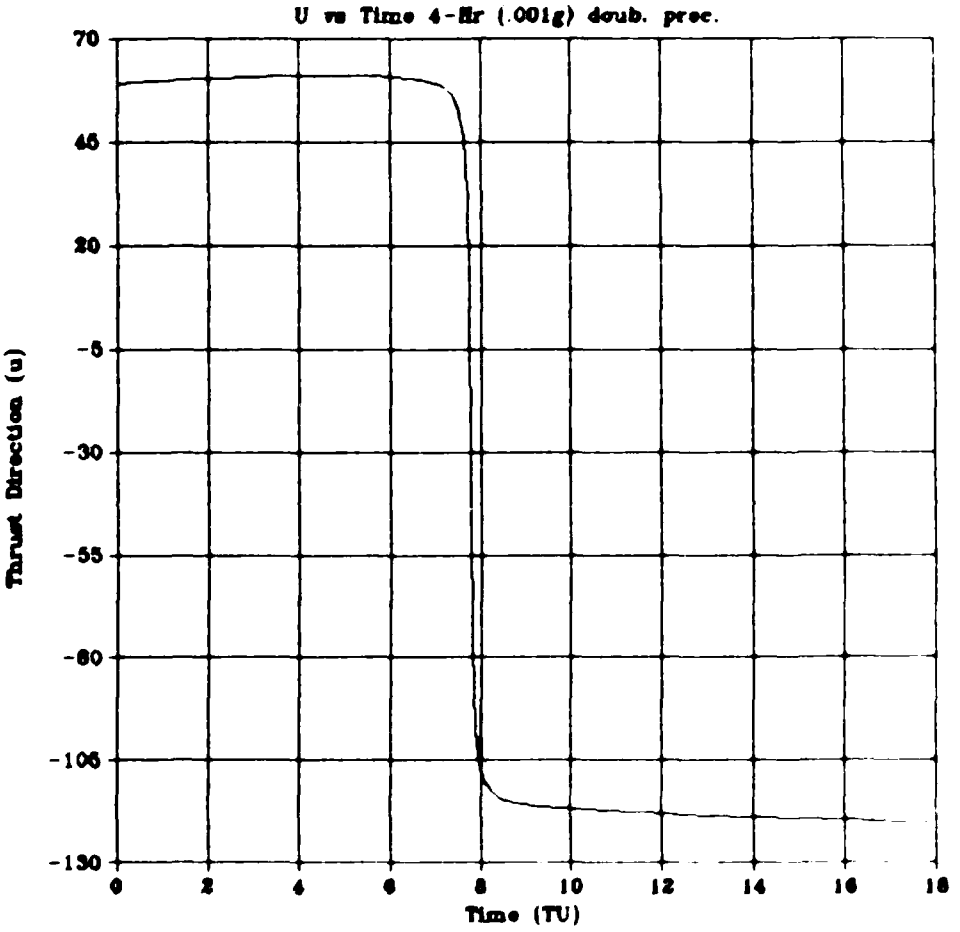


Figure 9.

THRUST DIRECTION (u) vs TIME -- .0005g_o Thrust -- 4 HR TOF

Double Precision, Single Satellite, Circular end Orbit

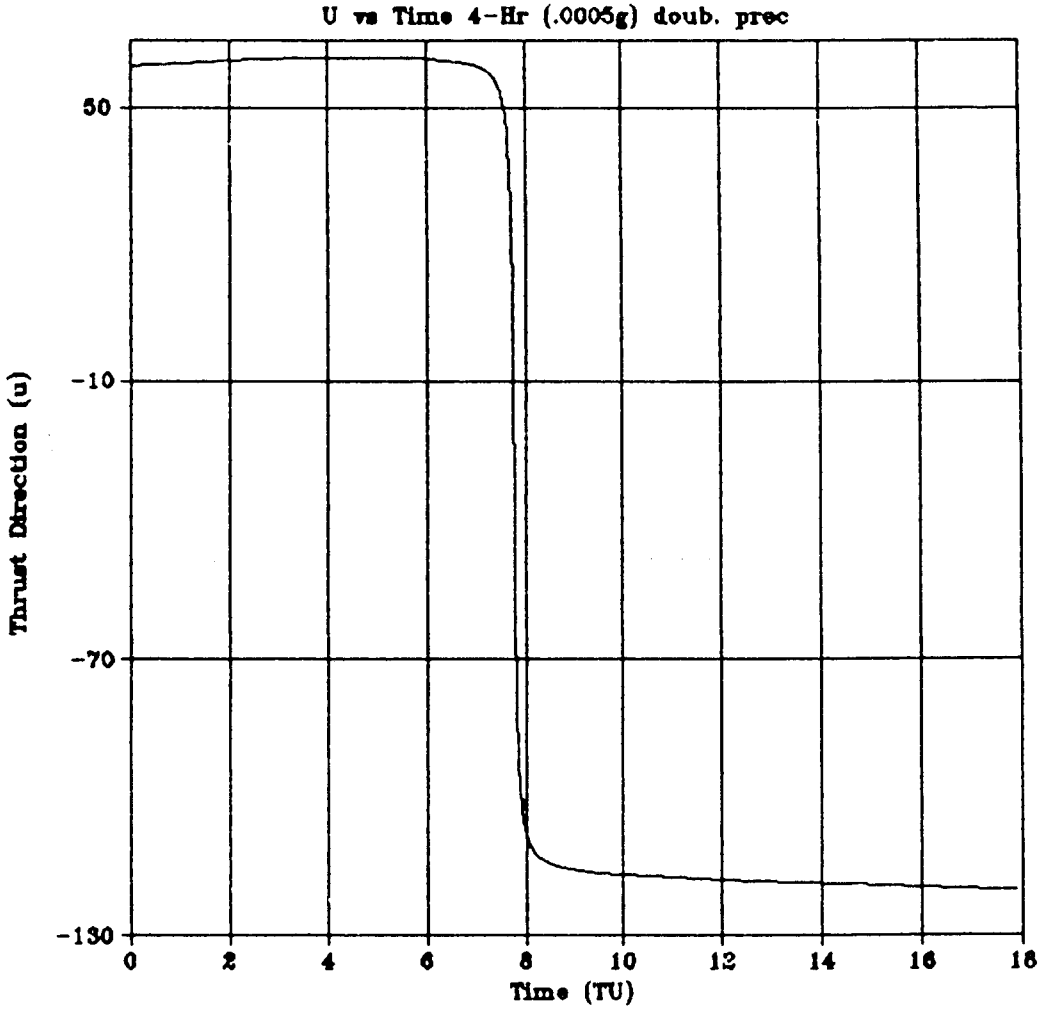


figure 10.

It appears that the rapid thrust direction change that occurs in all cases is dependent upon the total time of maneuvering. In the six hour maneuver times, this rapid thrust direction change goes through 180°; whereas in the four hour maneuver times, it goes through 0°. It becomes obvious that thrust direction history becomes dependent upon total thrust duration.

Circular Orbit Constraint at t_f *removed*

It became obvious by going through numerous output files, that much of the fuel being used to make the final orbit circular was in fact stealing from orbit radius increase. Since the objective was to maximize ρ (orbit radius), this constraint (equation 21) was removed in order to get a greater radius increase; therefore, increase the distance the target satellite was able to get away from the attacking satellite. In other words, this version utilizes the *same* amount of fuel to get significantly greater distances away from the threat. The boundary conditions that changed for this run were due to removing equation 20, 28, and 29; while including equations 51 – 53.

These results can be observed graphically in figures 11, 12, 13, and 14. From these figures, one observes that the thrust direction change now always goes through 0°. Also, there is no longer a rapid change from positive thrust direction angles to negative thrust direction angles. The four hour burns begin at approximately 27° of thrust direction angle and begin a moderately smooth transition (still highly non–linear) to their maximum negative values of about −120°. The six hour burns both begin at approximately 45° and transition to a maximum negative value of about −90°. Thrust direction angle is still seen as highly dependent upon the total burn time; the greater the total burn time, the greater the initial values of thrust direction angle.

THRUST DIRECTION (u) vs TIME -- .001g_o Thrust -- 6 HR TOF

Double Precision, Single Satellite, Non-Circular end Orbit

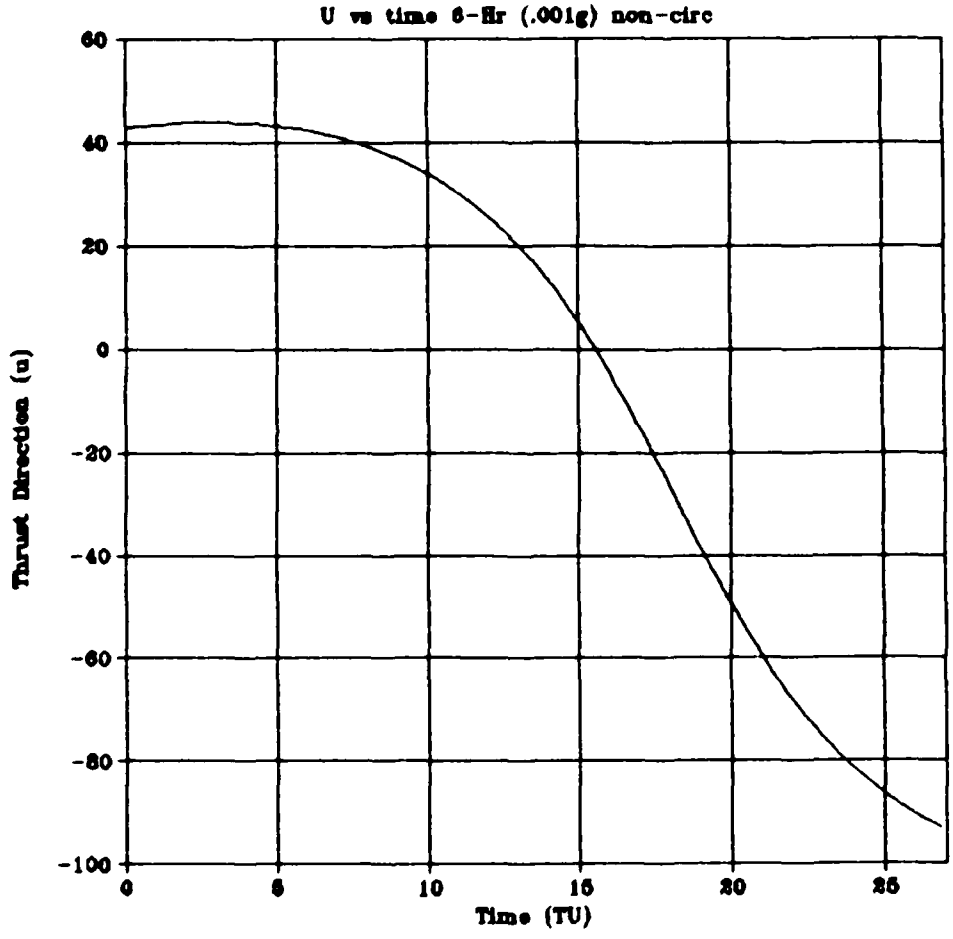


figure 11.

THRUST DIRECTION (u) vs TIME -- .0005g_o Thrust -- 6 HR TOF

Double Precision, Single Satellite, Non-Circular end Orbit

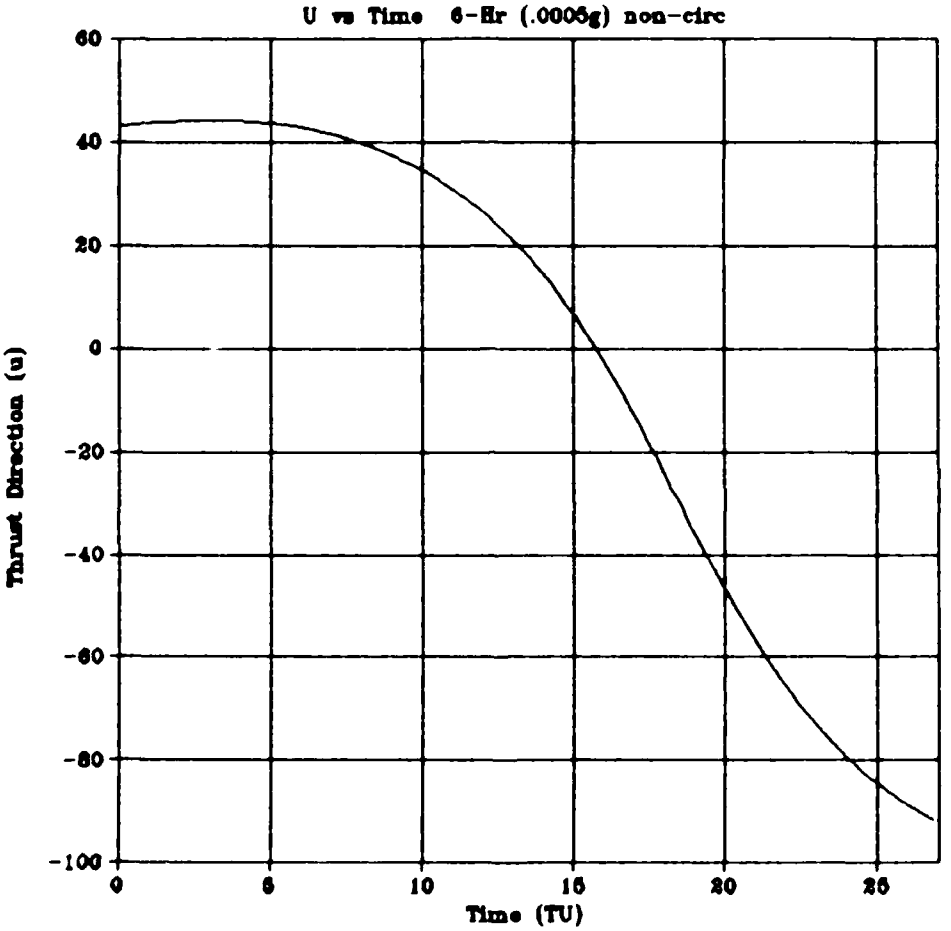


figure 13.

THRUST DIRECTION (u) vs TIME -- .001g_o Thrust -- 4 HR TOF

Double Precision, Single Satellite, Non-Circular end Orbit

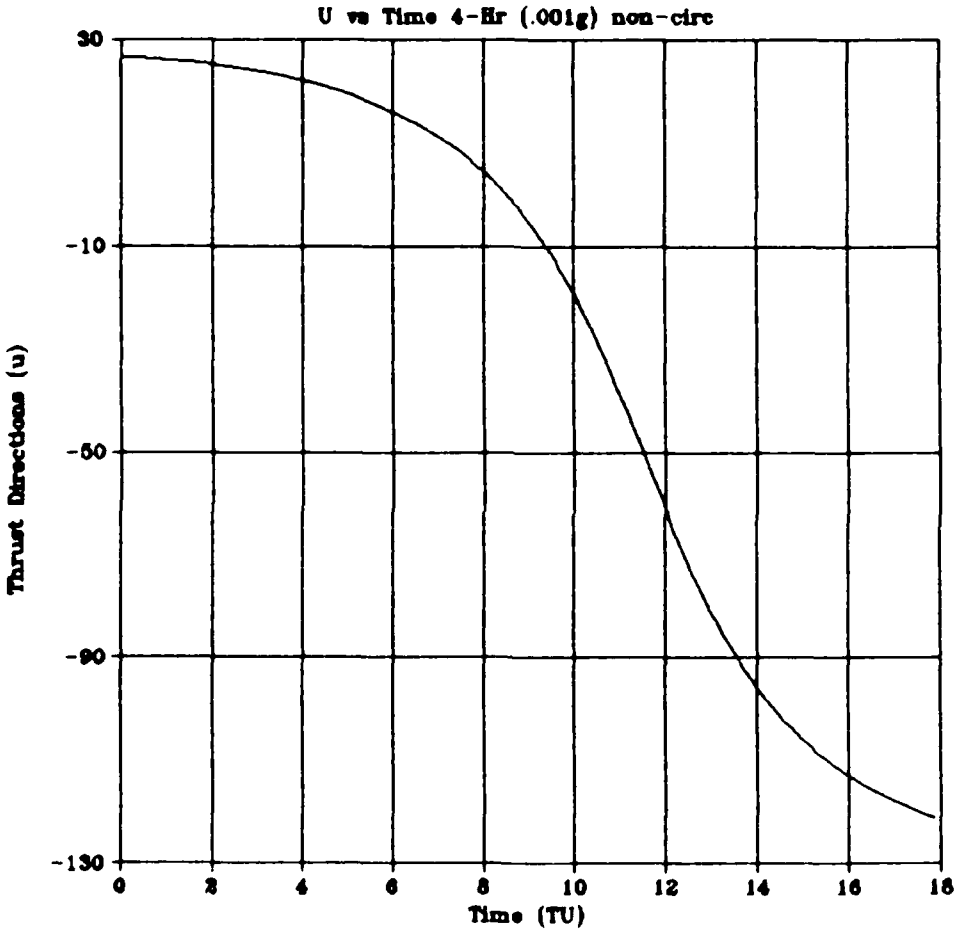


figure 13.

THRUST DIRECTION (u) vs TIME -- .0005g_o Thrust -- 4 HR TOF

Double Precision, Single Satellite, Non-Circular end Orbit

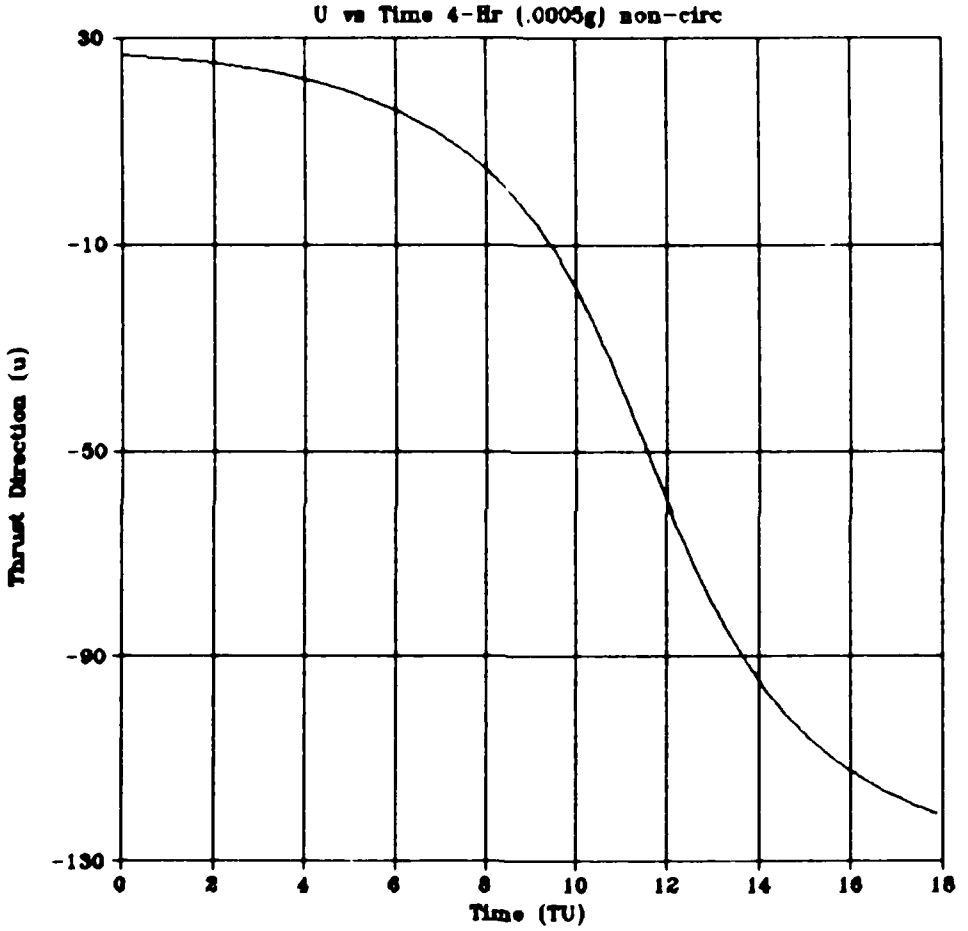


figure 14.

The significant part of this version is the distance increase. Orbit radius was increased ~1707 kilometers in the .001g_o six hour burn. This is 915 kilometers more than the previous version! The .0005g_o six hour burn increased radius by ~862 kilometers, almost 500 km more than that previously obtained. The four hour burns were equally impressive. The .001g_o four hour burn increased radius by ~665 kilometers; while the .0005g_o four hour burn increased the radius by ~335 kilometers.

Two Satellite Algorithm Results

The results from the two satellite portion of this algorithm are shown in figures 15, and 16. Once again, the only constraint on the two satellite problem is the tangency requirement of equation 20. Figure 15 shows both satellites beginning at the same point in space. The terminal boundary conditions are those of equations 20 and 51 – 53. One quarter of a geosynchronous orbit is shown as that of the attacking satellite. The target satellite is the one that has the ever increasing radius. It is interesting to note that not only is the radius increasing, but the period of its orbit is also changing ever so slightly. This ever so slight period increase has a tremendous effect on this problem.

The threat satellite's orbit was then gradually changed and the algorithm continually updated with previous output to keep a good "initialization grid" until the attacking satellite was in a very high eccentricity orbit with a radius of perigee of approximately 100 nautical miles and an apogee altitude equal to geosynchronous. The initial conditions eventually become those of equations 37 – 40. The algorithm was then repeated for both of the previous thrust magnitudes with both of the thrust burn times with the same terminal conditions as before.

Overall Satellite Depiction

***Two Satellites*: Both Satellites *initially* Geosynchronous**

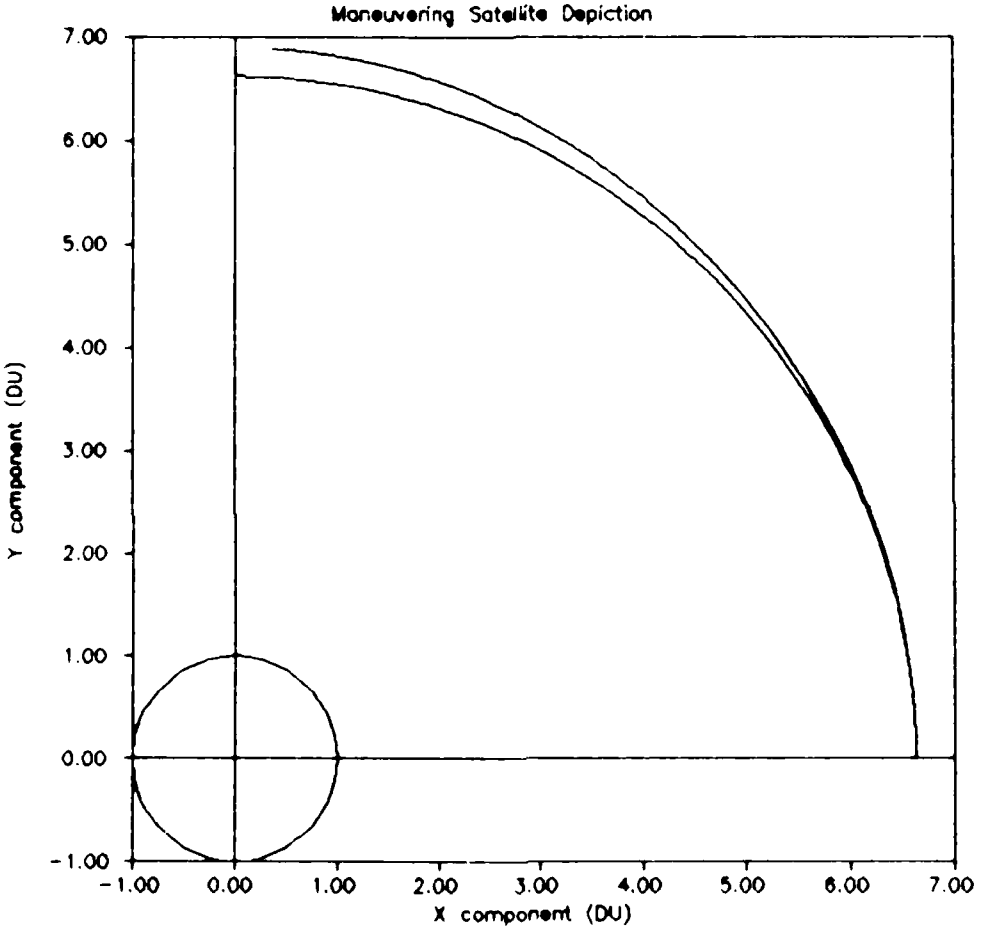


figure 15.

Overall Satellite Depiction

Two Satellites: **Target *initially* in Geosynchronous**

Attacker – High Eccentricity

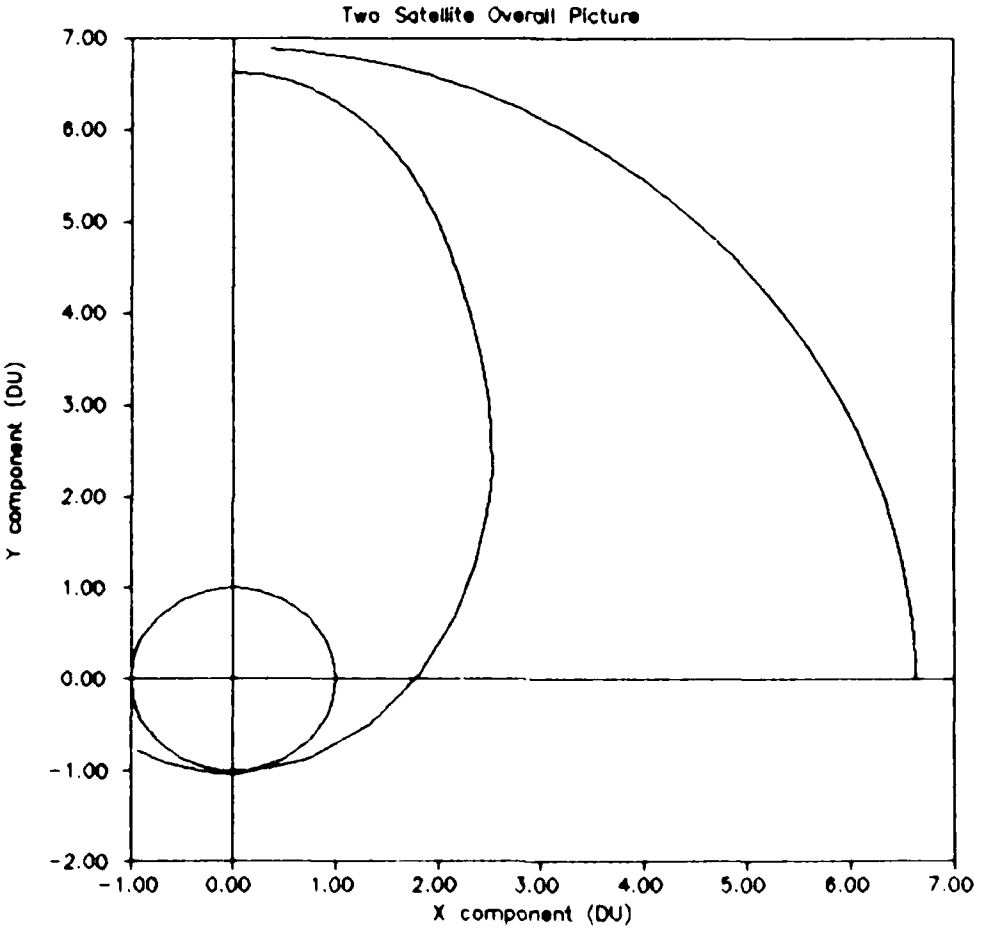


figure 16.

The results from this were nothing short of great! The biggest increase in distance came not only from the increase in the maneuvered satellite's radius, as was first assumed, but also from the fact that as the radius was increasing, the satellite was also slowing down in its orbit just enough to cause it to arrive at the intercept point area after the threat satellite had already left this area. The combined effect of increased period and increased radius greatly increased the total miss distance. (the distance between the attacker satellite and the target satellite) For the .001g_o thrust, six hour burn time, total distance between the two satellites at intercept time is now ~2800 kilometers. The .0005g_o thrust, six hour burn time, resulted in an ~1413 kilometers distance between the satellites. The .001g_o thrust, four hour burn time, gave an approximate 1000 kilometers and the .0005g_o thrust, four hour burn time, increased the distance just over 500 kilometers.

It is important to note that the thrust direction histories for the two satellite runs matched, exactly, the runs made previously with the fixed threat sphere and a single maneuvering satellite. Therefore, figures 11 - 14 are also valid representations of the two satellite versions of this algorithm

An example of the final two satellite transfer can be seen in figure 16 This figure shows that for a six hour burn time, the attacker is just before perigee in its orbit at the beginning of the burn. The two satellites continue in their respective orbits, with the target satellite continually changing its orbit. At the intercept time (six hours in this depiction), when the threat satellite is programmed to explode (or whatever), the target satellite is approximately 2800 kilometers further out and behind the threat, hopefully and apparently out of danger.

Minimum Time Problem

Now that the important part of the problem has been worked, it leaves only the problem of doing this problem in the optimum time. Section III tells how this algorithm was accomplished. A fixed orbit radius increase was selected, and the minimum time to get to that orbit radius was found. This problem was accomplished with both the $.001g_o$ thrust magnitude and the $.0005g_o$ thrust magnitude. The problem was first programmed for a single satellite and then attempted for the two satellite problem. A working model of the minimum time algorithm for the two satellite problem could not be achieved. Results achieved are for a single satellite with a fixed threat sphere. The tangency constraint remains the only constraint on this problem and the boundary conditions are included in the appendix for this algorithm. Figure 17 shows the thrust direction angle vs the parameterized time for the $.001g_o$ thrust magnitude case in which target radius was increased by 500 kilometers. Five hundred kilometers was used by Burk (8), but is not known to be sufficient for all possible attacks in space. The minimum time for a 500 km threat sphere was found to be 12.0085 TU or 2 hours, 41 minutes, 28.6 seconds. Target thrust direction angle begins at about 21° and transitions fairly smoothly toward a maximum negative value of about $-128°$.

Figure 18 shows the thrust direction angle vs the parameterized time for the $.0005g_o$ thrust magnitude case in which target radius was also increased by 500 kilometers. (500 km threat sphere) The minimum time in this case was found to be 16.7495 TU or 3 hours, 45 minutes, 13.7 seconds. Here the target thrust direction angle begins just above 27° and proceeds to a maximum negative value of $-107°$.

A run was also made with .001go thrust magnitude in which target radius was increased by 1000 km. The minimum time for this case was 16.7532 TU or 3 hours, 45 minutes, 16.7 seconds. In other words, with twice the thrust magnitude, it was possible to increase the radius twice as far in almost the same time. This proved to be merely coincidental however, as other runs did not prove this linear relationship out. For a 1500 km orbit radius increase with .001go thrust magnitude, the time was 25.3598 TU or 5 hours, 41 minutes, 0.6 seconds.

The minimization of the two satellite problem was attempted to incorporate all aspects of the original problem statement; that being to get the target satellite as far away from the attacking satellite in the minimum amount of time. The attacking satellite's orbital dynamics coupled with the complex dynamics of the minization problem proved to be too much for the DVCPR routine. Figure 19, however, shows how this procedure would most likely appear. In this figure, The attacking satellite is actually just above the X axis, on its way to its apogee altitude which will be coincident with the target satellite at geosynchronous altitude. The target satellite; however, begins its thrust maneuver, and a *few short hours* later is presumedly safely away from the threat instead of coincident with it.

Minimum Time Problem

.001g_o Thrust -- 500 km radius increase

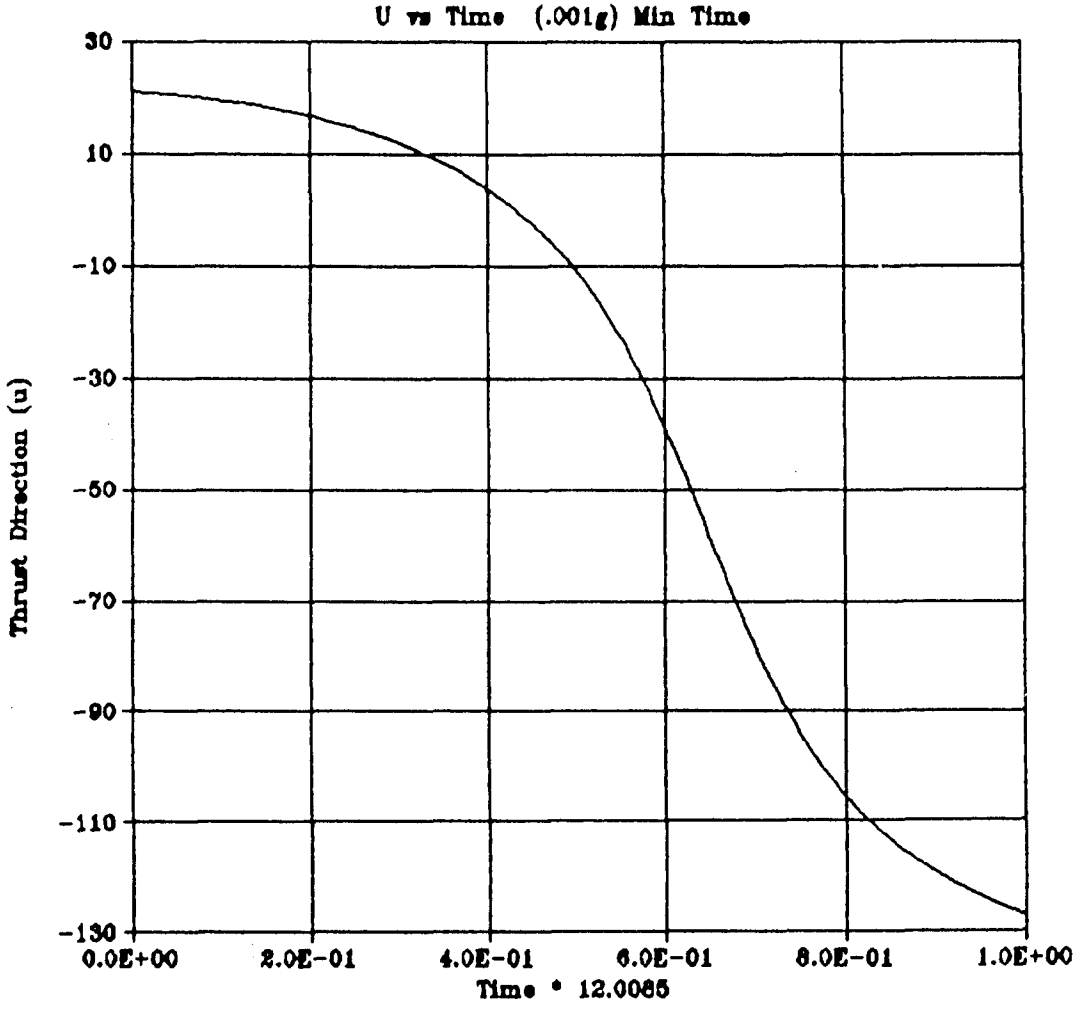


figure 17.

Minimum Time Problem

.0005g_o Thrust -- 500 km radius increase

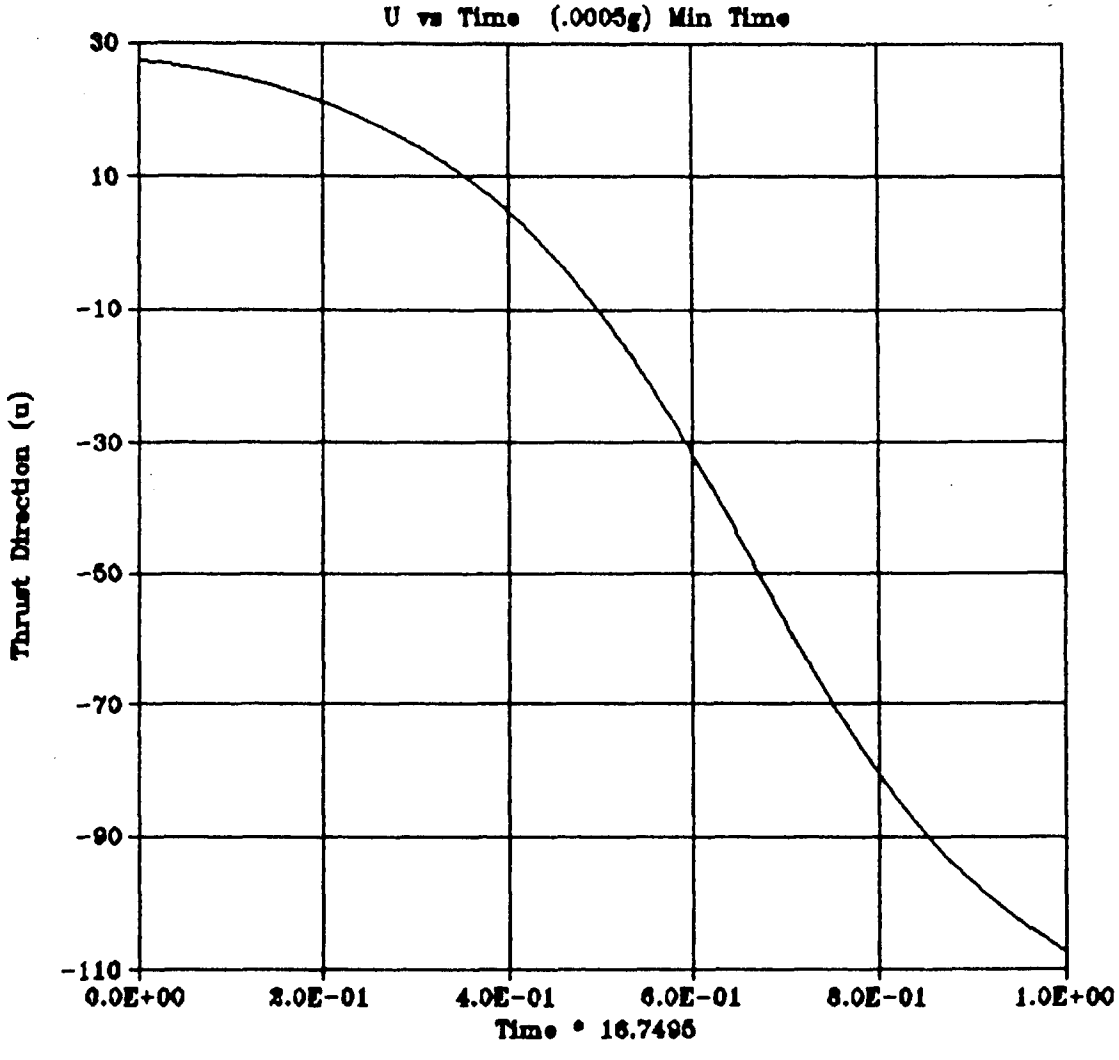


figure 18.

Proposed Minimum Time Problem

.001g_o Thrust -- 1000 km threat distance

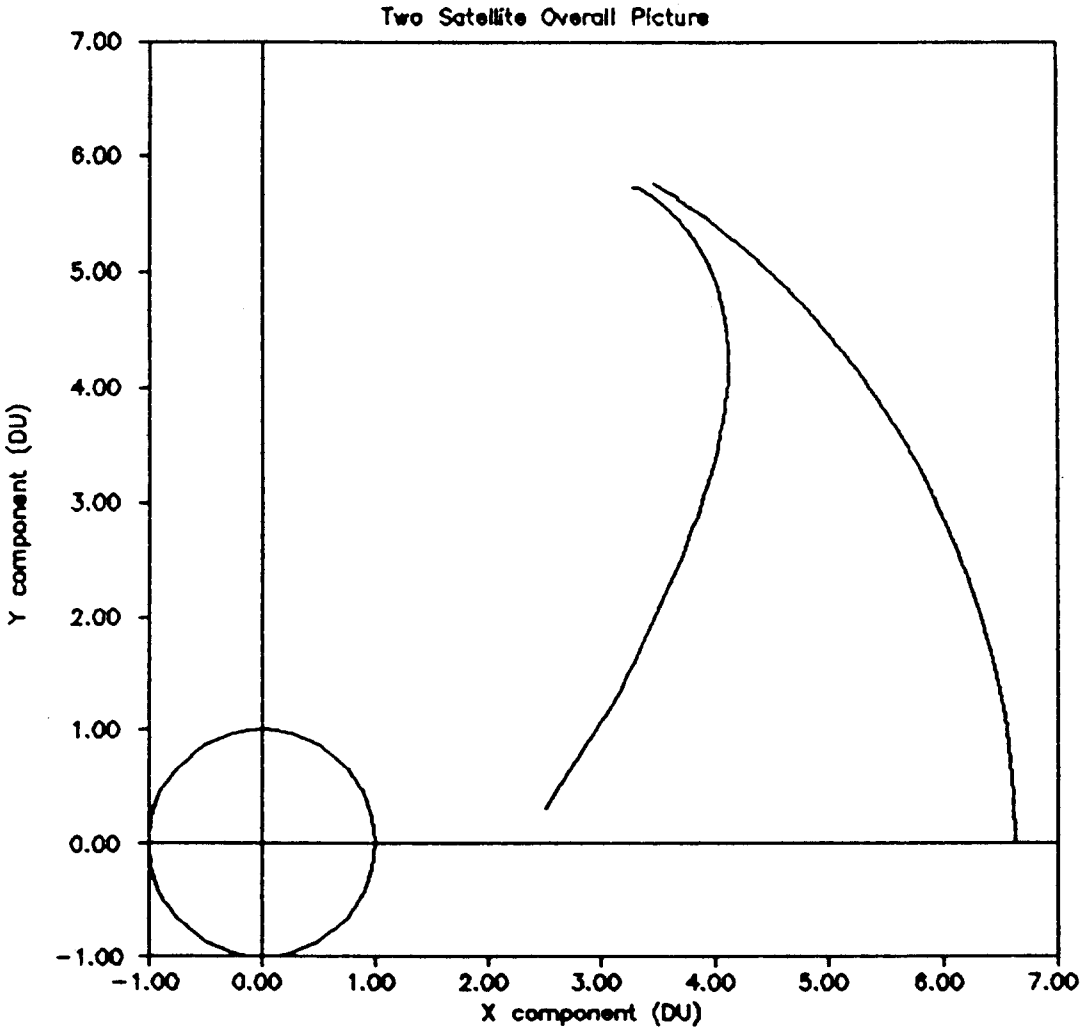


figure 19.

V. Conclusions, Tactics, and Suggestions

Conclusions

The problem of optimal time transfer to some desired terminal radius increase using a constant magnitude, low thrust propulsion system has been investigated.

Numerical solutions to this highly non-linear TPBVP were attainable through the IMSL routine DVCPR. Although this routine must be initialized (rather accurately) in order to find a converged solution, it seems that it would be applicable to a wide class of orbital transfer problems.

For geosynchronous orbits, it is now determinable that the radius increase has a non-linear relationship that is proportional to the burn time. It is obvious to say that the longer the burn time, the greater the orbit radius will be; however, it becomes significant that this happens very rapidly between the four and six hour points. Over one thousand kilometers is added to the orbit radius between these times in the .001g_0 thrust magnitude case. In the .0005g_0 thrust magnitude case, over five hundred kilometers is added to the radius between the four and six hour points.

Initial thrust direction angle also is greatly dependent upon the total time in question. The greater the total burn time, the greater the starting value of the thrust direction. For the minimum time cases, this does not appear to be true however. It appears that the smaller the thrust, the greater the beginning value for thrust direction.

The key conclusion however is that a constant, low thrust propulsion system does seem to be a valid alternative to the large impulsive propulsion system studies in the current literature in accomplishing evasive orbital

maneuvers. In view of the accuracy of these results, this approach forms the basis of an effective numerical technique for completely general optimal low thrust orbital transfer.

Tactics

Many practical aspects of the attack avoidance problem have not been dealt with in this thesis. In a real attack on a satellite, the threat would have to be detected and then tracked for a while to determine its orbit. From this information the defender could deduce the intended target and the time of intercept. The defender would have to come to some conclusion regarding the lethal radius of the threat, either from intelligence information or from "lucky" guessing. He might decide to defeat the threat by evasive maneuvering (as this thesis discusses), by attacking the attacker, by attacking the enemy's control system or communications, by using chaff or decoys, or by relying on his satellite's hardening. He could also decide to do nothing at all and just bear the loss of a satellite.

As was stated earlier, the earlier the maneuver begins, the more distance away from the attacker the satellite can achieve. The attacker may detect this maneuver, and re-establish the intercept by a countermaneuver on its part. This would then have to be detected and another evasive maneuver performed by the target. Depending upon available fuel onboard, this could be a losing effort. This all becomes a type of cat and mouse game, with each player attempting to out guess his opponent. Under these circumstances, the orbital evasion problem becomes the classical military game of trying to outfox the enemy.

Suggestions for Further Work

There are many other areas of the orbital evasion problem that could be addressed, but four more areas of the problem as addressed in this thesis will be mentioned. They are as follows: the performance of the algorithm, the effects of uncertainty, the accuracy of the model, and the development of a three dimensional model.

This algorithm proved to give good results, but only after much frustration and time was spent in finding good initial values in which to begin the problem. A method should be found to make the algorithm reliable without having to initialize the grid for the Newton algorithm. One alternative approach to this may be to have a library of example data, for various types of orbits and maneuvers. It was found, that once a fairly good set of data was available as a starting guess, the algorithm achieved superb results.

The algorithm in this thesis does not account for uncertainties in the input data. How is it determined that a specific satellite is under attack? With the increasing density of satellites in geosynchronous orbits, it will remain very hard to establish which satellite (or satellites) needs to be maneuvered. Once this is determined, can the required thrust magnitude or more importantly the required thrust direction angle be programmed into the satellite accurately? Since the thrust direction angle is constantly changing, some means of communicating with the satellite must be maintained throughout the maneuver or pre-programmed into the satellite guidance system.

This algorithm does not take into account any perturbation effects that effect all orbiting bodies, nor does it account for a maneuvering threat or a non-spherical threat volume. Perturbation algorithms are available but some

are classified and others can not be implemented with available resources. It has, however, been observed in the open literature that a maneuvering threat is currently beyond the scope of what our current enemies capabilities are. Also, any other shaped threat volume, would have to be addressed in a separate algorithm.

The last topic for further work is obviously a three dimensional model. This thesis dealt only with geosynchronous orbits and co-planer attacking satellites. In order to expand this algorithm to attacking satellites in inclined orbits, a three dimensional model must be developed. This model may also address the problem of returning the satellite to its mission orbit in an optimal or minimum fuel type of burn.

Appendix

Computer listings of *DVCPR* subroutines **FCNI, FCNJ, and FCNB**

Single Satellite Version:

```
      SUBROUTINE FCNI(N,T,X,XP)
      DOUBLE PRECISION T,X(N),XP(N),MU,RO,B,D,TH,M0,MD,U
      INTEGER N
      COMMON /A/ MU /B/ TH,M0,MD
      RO=DSQRT(X(1)**2+X(2)**2)
      IF (RO .EQ. 0) RO=1.0
      B=TH/(M0-MD*T)
      U=DATAN2(X(8),X(7))
      D=X(7)*X(1) + X(8)*X(2)
      XP(1)=X(3)
      XP(2)=X(4)
      XP(3)=-MU*X(1)/RO**3 + B*DCOS(U)
      XP(4)=-MU*X(2)/RO**3 + B*DSIN(U)
      XP(5)=MU*(X(7)/RO**3 - 3.0d+00*X(1)*D/RO**5)
      XP(6)=MU*(X(8)/RO**3 - 3.0d+00*X(2)*D/RO**5)
      XP(7)=-X(5)
      XP(8)=-X(6)
      RETURN
      END
*
*****************************************************************
*
      SUBROUTINE FCNJ(N,T,X,PD)
      DOUBLE PRECISION T,X(N),PD(N,N),MU,RO,B,C,D,TH,M0,
     &  MD,E,F,G,H,U
      INTEGER N
      COMMON /A/ MU /B/ TH,M0,MD
      RO=DSQRT(X(1)**2+X(2)**2)
      IF (RO .EQ. 0) RO=1.0
      B=TH/(M0-MD*T)
      U=DATAN2(X(8),X(7))
      F=X(8)/X(7)
      C=(1.0d+00 + F**2)
      D=X(7)*X(1) + X(8)*X(2)
      E=X(8)*X(1) + X(7)*X(2)
      G=3.0d+00*X(7)*X(1) + X(8)*X(2)
      H=3.0d+00*X(2)*X(8) + X(1)*X(7)
```

```
      PD(1,1)=0.0
      PD(1,2)=0.0
      PD(1,3)=1.0d+00
      PD(1,4)=0.0
      PD(1,5)=0.0
      PD(1,6)=0.0
      PD(1,7)=0.0
      PD(1,8)=0.0

      PD(2,1)=0.0
      PD(2,2)=0.0
      PD(2,3)=0.0
      PD(2,4)=1.0d+00
      PD(2,5)=0.0
      PD(2,6)=0.0
      PD(2,7)=0.0
      PD(2,8)=0.0

      PD(3,1)=MU*(-1.0d+00/RO**3 + 3.0d+00*X(1)**2/RO**5)
      PD(3,2)=3.0d+00*MU*X(1)*X(2)/RO**5
      PD(3,3)=0.0
      PD(3,4)=0.0
      PD(3,5)=0.0
      PD(3,6)=0.0
      PD(3,7)=B*DSIN(U)*X(8)/(C*X(7)**2)
      PD(3,8)=-B*DSIN(U)/(C*X(7))

      PD(4,1)=PD(3,2)
      PD(4,2)=MU*(-1.0d+00/RO**3 + 3.0d+00*X(2)**2/RO**5)
      PD(4,3)=0.0
      PD(4,4)=0.0
      PD(4,5)=0.0
      PD(4,6)=0.0
      PD(4,7)=-B*DCOS(U)*X(8)/(C*X(7)**2)
      PD(4,8)=B*DCOS(U)/(C*X(7))

      PD(5,1)=MU*(-3.0d+00*G/RO**5 + 15.0d+00*X(1)**2*D
     $     /RO**7)
      PD(5,2)=MU*(-3.0d+00*E/RO**5 + 15.0d+00*X(1)*X(2)*D
     $     /RO**7)
      PD(5,3)=0.0
      PD(5,4)=0.0
      PD(5,5)=0.0
      PD(5,6)=0.0
      PD(5,7)=MU*(1.0d+00/RO**3 - 3.0d+00*X(1)**2/RO**5)
      PD(5,8)=-3.0d+00*MU*X(1)*X(2)/RO**5
```

```
      PD(6,1)=PD(5,2)
      PD(6,2)=MU*(-3.0d+00*H/RO**5 + 15.0d+00*X(2)**2*D
     &          /RO**7)
      PD(6,3)=0.0
      PD(6,4)=0.0
      PD(6,5)=0.0
      PD(6,6)=0.0
      PD(6,7)=PD(5,8)
      PD(6,8)=MU*(1.0d+00/RO**3 - 3.0d+00*X(2)**2/RO**5)

      PD(7,1)=0.0
      PD(7,2)=0.0
      PD(7,3)=0.0
      PD(7,4)=0.0
      PD(7,5)=-1.0d+00
      PD(7,6)=0.0
      PD(7,7)=0.0
      PD(7,8)=0.0

      PD(8,1)=0.0
      PD(8,2)=0.0
      PD(8,3)=0.0
      PD(8,4)=0.0
      PD(8,5)=0.0
      PD(8,6)=-1.0d+00
      PD(8,7)=0.0
      PD(8,8)=0.0

      RETURN
      END
*
*****************************************************************
*
      SUBROUTINE FCNB(N,XA,XB,F)
      DOUBLE PRECISION XA(N),XB(N),F(N),KS,KM,RB
      INTEGER N
      COMMON /CONV/ KM,KS
      RB=DSQRT(XB(1)**2+XB(2)**2)
      F(1)=XA(1) - KM
      F(2)=XA(2)
      F(3)=XA(3)
      F(4)=XA(4) - KS
      F(5)=XB(1)*XB(3)+XB(2)*XB(4)
      F(6)=XB(7)*XB(3)+XB(8)*XB(4)
      F(7)=XB(5)*XB(1)+XB(6)*XB(2) - RB
      F(8)=XB(8)-XB(7)*XB(2)/XB(1)
      RETURN
      END
*
*****************************************************************
```

Two Satellite Version

```
      SUBROUTINE FCNI(N,T,X,XP)
      DOUBLE PRECISION T,X(N),XP(N),MU,B,D,TH,M0,MD,R1(4),
     $ R2(4),R21,R22,R23,R24,V21,V22,V23,V24,R20(4),V20(4),
     $ C,E,F
      INTEGER N
      COMMON /A/ MU /B/ TH,M0,MD
      COMMON /RS/ R21,R22,R23,R24,V21,V22,V23,V24
      R20(1)=R21
      R20(2)=R22
      R20(3)=R23
      R20(4)=R24
      V20(1)=V21
      V20(2)=V22
      V20(3)=V23
      V20(4)=V24
      CALL KEPLER(T,MU,R20,V20,R2,V2)
      R1(1)=R2(1)-X(1)
      R1(2)=R2(2)-X(2)
      R1(3)=0.0
      B=TH/(M0-MD*T)
      C=X(7)*R1(1) + X(8)*R1(2)
      D=DSQRT(R1(1)**2+R1(2)**2+R1(3)**2)
      E=DSQRT(R2(1)**2+R2(2)**2+R2(3)**2)
      F=DSQRT(X(7)**2+X(8)**2)
      XP(1)=X(3)
      XP(2)=X(4)
      XP(3)=-MU*R2(1)/E**3 + MU*R1(1)/D**3 - B*X(7)/F
      XP(4)=-MU*R2(2)/E**3 + MU*R1(2)/D**3 - B*X(8)/F
      XP(5)=MU*(X(7)/D**3 - 3.0d+00*R1(1)*C/D**5)
      XP(6)=MU*(X(8)/D**3 - 3.0d+00*R1(2)*C/D**5)
      XP(7)=-X(5)
      XP(8)=-X(6)
      RETURN
      END
*
*****************************************************************
*
      SUBROUTINE FCNJ(N,T,X,PD)
      DOUBLE PRECISION T,X(N),PD(N,N),MU,B,C,D,TH,M0,MD,E,
     $  F,G,R2(4),R21,R22,R23,R24,V21,V22,V23,V24,R20(4),
     $  V20(4),R1(4)
      INTEGER N
      COMMON /A/ MU /B/ TH,M0,MD
      COMMON /RS/ R21,R22,R23,R24,V21,V22,V23,V24
      R20(1)=R21
      R20(2)=R22
      R20(3)=R23
      R20(4)=R24
      V20(1)=V21
      V20(2)=V22
      V20(3)=V23
```

```
V20(4)=V24
CALL KEPLER(T,MU,R20,V20,R2,V2)
R1(1)=R2(1)-X(1)
R1(2)=R2(2)-X(2)
R1(3)=0.0
B=TH/(M0-MD*T)
C=X(7)*R1(1) + X(8)*R1(2)
D=DSQRT(R1(1)**2+R1(2)**2+R1(3)**2)
E=DSQRT(R2(1)**2+R2(2)**2+R2(3)**2)
F=DSQRT(X(7)**2+X(8)**2)
G=X(7)*R1(2) + X(8)*R1(1)

PD(1,1)=0.0
PD(1,2)=0.0
PD(1,3)=1.0d+00
PD(1,4)=0.0
PD(1,5)=0.0
PD(1,6)=0.0
PD(1,7)=0.0
PD(1,8)=0.0

PD(2,1)=0.0
PD(2,2)=0.0
PD(2,3)=0.0
PD(2,4)=1.0d+00
PD(2,5)=0.0
PD(2,6)=0.0
PD(2,7)=0.0
PD(2,8)=0.0

PD(3,1)=MU*(-1.0d+00/D**3 + 3.0d+00*R1(1)**2/D**5)
PD(3,2)=3.0d+00*MU*R1(1)*R1(2)/D**5
PD(3,3)=0.0
PD(3,4)=0.0
PD(3,5)=0.0
PD(3,6)=0.0
PD(3,7)= -B/F - B*X(7)**2/F**3
PD(3,8)=-B*X(7)*X(8)/F**3

PD(4,1)=PD(3,2)
PD(4,2)=MU*(-1.0d+00/D**3 + 3.0d+00*R1(2)**2/D**5)
PD(4,3)=0.0
PD(4,4)=0.0
PD(4,5)=0.0
PD(4,6)=0.0
PD(4,7)=PD(3,8)
PD(4,8)= -B/F - B*X(8)**2/F**3
```

```
      PD(5,1)=MU*((6.0d+00*X(7)*R1(1) + 3.0d+00*C)/D**5
     $           15.0d+00*R1(1)**2*C/D**7)
      PD(5,2)=MU*(3.0d+00*G/D**5 + 15.0d+00*R1(1)*R1(2)*C
     $          /D**7)
      PD(5,3)=0.0
      PD(5,4)=0.0
      PD(5,5)=0.0
      PD(5,6)=0.0
      PD(5,7)=MU*(1.0d+00/D**3 - 3.0d+00*R1(1)**2/D**5)
      PD(5,8)=-3.0d+00*MU*R1(1)*R1(2)/D**5

      PD(6,1)=PD(5,2)
      PD(6,2)=MU*((6.0d+00*X(8)*R1(2) + 3.0d+00*C)/D**5
     $          - 15.0d+00*R1(2)**2*C/D**7)
      PD(6,3)=0.0
      PD(6,4)=0.0
      PD(6,5)=0.0
      PD(6,6)=0.0
      PD(6,7)=PD(5,8)
      PD(6,8)=MU*(1.0d+00/D**3 - 3.0d+00*R1(2)**2/D**5)

      PD(7,1)=0.0
      PD(7,2)=0.0
      PD(7,3)=0.0
      PD(7,4)=0.0
      PD(7,5)=-1.0d+00
      PD(7,6)=0.0
      PD(7,7)=0.0
      PD(7,8)=0.0

      PD(8,1)=0.0
      PD(8,2)=0.0
      PD(8,3)=0.0
      PD(8,4)=0.0
      PD(8,5)=0.0
      PD(8,6)=-1.0d+00
      PD(8,7)=0.0
      PD(8,8)=0.0

      RETURN
      END
```

```
*****************************************************************
*
      SUBROUTINE FCNB(N,XA,XB,F)
      DOUBLE PRECISION XA(N),XB(N),F(N),KS,KM,RB
      INTEGER N
      COMMON /CONV/ KM,KS
      RB=DSQRT(XB(1)**2+XB(2)**2)
      F(1)=XA(1) + 7.552551d+00
      F(2)=XA(2) + .7744873d+00
      F(3)=XA(3) - 1.0241614d+00
      F(4)=XA(4) + .9642712d+00
      F(5)=XB(1)*XB(3)+XB(2)*XB(4)
      F(6)=XB(8)-XB(7)*XB(2)/XB(1)
      F(7)=XB(3)*XB(7)+XB(4)*XB(8)
      F(8)=XB(5)*XB(1)+XB(6)*XB(2)-RB
      RETURN
      END
*****************************************************************
```

Time Minimization Version

```
      SUBROUTINE FCNI(N,T,X,XP)
      DOUBLE PRECISION T,X(N),XP(N),MU,RO,B,D,TH,M0,MD,U
      INTEGER N
      COMMON /A/ MU /B/ TH,M0,MD
      RO=DSQRT(X(1)**2+X(2)**2)
      IF (RO .EQ. 0) RO=1.0
      B=TH/(M0-MD*T)
      U=DATAN2(X(8),X(7))
      D=X(7)*X(1) + X(8)*X(2)
      XP(1)=X(3)*X(9)
      XP(2)=X(4)*X(9)
      XP(3)=(-MU*X(1)/RO**3 + B*DCOS(U))*X(9)
      XP(4)=(-MU*X(2)/RO**3 + B*DSIN(U))*X(9)
      XP(5)=MU*(X(7)/RO**3 - 3.0d+00*X(1)*D/RO**5)*X(9)
      XP(6)=MU*(X(8)/RO**3 - 3.0d+00*X(2)*D/RO**5)*X(9)
      XP(7)=-X(5)*X(9)
      XP(8)=-X(6)*X(9)
      XP(9)=0.0
      RETURN
      END
*
```

```
C*********************************************************************
C
      SUBROUTINE FCNJ(N,T,X,PD)
      DOUBLE PRECISION  T,X(N),PD(N,N),MU,RO,B,C,D,TH,M0,
     &  MD,E, F,G,H,U
      INTEGER N
      COMMON /A/ MU /B/ TH,M0,MD
      RO=DSQRT(X(1)**2+X(2)**2)
      IF (RO .EQ. 0) RO=1.0
      B=TH/(M0-MD*T)
      U=DATAN2(X(8),X(7))
      F=X(8)/X(7)
      C=(1.0d+00 + F**2)
      D=X(7)*X(1) + X(8)*X(2)
      E=X(8)*X(1) + X(7)*X(2)
      G=3.0d+00*X(7)*X(1) + X(8)*X(2)
      H=3.0d+00*X(2)*X(8) + X(1)*X(7)

      PD(1,1)=0.0
      PD(1,2)=0.0
      PD(1,3)=X(9)
      PD(1,4)=0.0
      PD(1,5)=0.0
      PD(1,6)=0.0
      PD(1,7)=0.0
      PD(1,8)=0.0
      PD(1,9)=X(3)

      PD(2,1)=0.0
      PD(2,2)=0.0
      PD(2,3)=0.0
      PD(2,4)=X(9)
      PD(2,5)=0.0
      PD(2,6)=0.0
      PD(2,7)=0.0
      PD(2,8)=0.0
      PD(2,9)=X(4)

      PD(3,1)=MU*(-1.0d+00/RO**3 + 3.0d+00*X(1)**2
     &     /RO**5)*X(9)
      PD(3,2)=3.0d+00*MU*X(1)*X(2)/RO**5*X(9)
      PD(3,3)=0.0
      PD(3,4)=0.0
      PD(3,5)=0.0
      PD(3,6)=0.0
      PD(3,7)=B*DSIN(U)*X(8)/(C*X(7)**2)*X(9)
      PD(3,8)=-B*DSIN(U)/(C*X(7))*X(9)
      PD(3,9)=-MU*X(1)/RO**3 + B*DCOS(U)
```

```
      PD(4,1)=PD(3,2)
      PD(4,2)=MU*(-1.0d+00/RO**3 + 3.0d+00*X(2)**2
     $    /RO**5)*X(9)
      PD(4,3)=0.0
      PD(4,4)=0.0
      PD(4,5)=0.0
      PD(4,6)=0.0
      PD(4,7)=-B*DCOS(U)*X(8)/(C*X(7)**2)*X(9)
      PD(4,8)=B*DCOS(U)/(C*X(7))*X(9)
      PD(4,9)=-MU*X(2)/RO**3 + B*DSIN(U)

      PD(5,1)=MU*(-3.0d+00*G/RO**5+15.0d+00*X(1)**2*D
     $    /RO**7)*X(9)
      PD(5,2)=MU*(-3.0d+00*E/RO**5+15.0d+00*X(1)*X(2)*D
     $    /RO**7)*X(9)
      PD(5,3)=0.0
      PD(5,4)=0.0
      PD(5,5)=0.0
      PD(5,6)=0.0
      PD(5,7)=MU*(1.0d+00/RO**3 - 3.0d+00*X(1)**2
     $    /RO**5)*X(9)
      PD(5,8)=-3.0d+00*MU*X(1)*X(2)/RO**5 * X(9)
      PD(5,9)=MU*(X(7)/RO**3 - 3.0d+00*X(1)*D/RO**5)

      PD(6,1)=PD(5,2)
      PD(6,2)=MU*(-3.0d+00*H/RO**5+15.0d+00*X(2)**2*D
     $    /RO**7)*X(9)
      PD(6,3)=0.0
      PD(6,4)=0.0
      PD(6,5)=0.0
      PD(6,6)=0.0
      PD(6,7)=PD(5,8)
      PD(6,8)=MU*(1.0d+00/RO**3 - 3.0d+00*X(2)**2
     $    /RO**5)*X(9)
      PD(6,9)=MU*(X(8)/RO**3 - 3.0d+00*X(2)*D/RO**5)

      PD(7,1)=0.0
      PD(7,2)=0.0
      PD(7,3)=0.0
      PD(7,4)=0.0
      PD(7,5)=-X(9)
      PD(7,6)=0.0
      PD(7,7)=0.0
      PD(7,8)=0.0
      PD(7,9)=-X(5)
```

```
      PD(8,1)=0.0
      PD(8,2)=0.0
      PD(8,3)=0.0
      PD(8,4)=0.0
      PD(8,5)=0.0
      PD(8,6)=-X(9)
      PD(8,7)=0.0
      PD(8,8)=0.0
      PD(8,9)=-X(8)

      PD(9,1)=0.0
      PD(9,2)=0.0
      PD(9,3)=0.0
      PD(9,4)=0.0
      PD(9,5)=0.0
      PD(9,6)=0.0
      PD(9,7)=0.0
      PD(9,8)=0.0
      PD(9,9)=0.0

      RETURN
      END
C
C*****************************************************************
C
      SUBROUTINE FCNB(N,XA,XB,F)
      DOUBLE PRECISION XA(N),XB(N),F(N),KS,KM,RB,U,B,XP3,
     &  XP4,TH,M0,MD,MU
      INTEGER N
      COMMON /A/ MU /CONV/ KM,KS /B/ TH,M0,MD
      RB=DSQRT(XB(1)**2+XB(2)**2)
      U=DATAN2(XB(8),XB(7))
      B=TH/(M0-MD*XB(9))
      XP3=XB(9)*XB(7)*(-MU*XB(1)/RB**3+B*DCOS(U))
      XP4=XB(9)*XB(8)*(-MU*XB(2)/RB**3+B*DSIN(U))
      F(1)=XA(1) - KM
      F(2)=XA(2)
      F(3)=XA(3)
      F(4)=XA(4) - KS
      F(5)=XB(1)*XB(3)+XB(2)*XB(4)
      F(6)=XB(7)*XB(3)+XB(8)*XB(4)
      F(7)=RB - 6.70118388d+00
      F(8)=XB(5)*XB(1)+XB(6)*XB(2) - RB
      F(9)=XB(9)+XB(9)*(XB(5)*XB(3)+XB(6)*XB(4))+XP3+XP4
      RETURN
      END
C*****************************************************************
```

Computer listing of Kepler Orbit Determiner

```
      SUBROUTINE KEPLER(T,MU,R0,V0,R,V)
      DOUBLE PRECISION T,R0(4),V0(4),R(4),V(4),E,MU,
     $   SMU,A,X0,XN,Z,SZ,C,S,TN,DT,DOT,TOL,F,G,FD,GD
      if (T .eq. 0) then
         R(1)=R0(1)
         R(2)=R0(2)
         R(3)=R0(3)
         V(1)=V0(1)
         V(2)=V0(2)
         V(3)=V0(3)
        R(4)=DSQRT(R(1)**2+R(2)**2+R(3)**2)
        RETURN
      endif
      DOT=R0(1)*V0(1)+R0(2)*V0(2)+R0(3)*V0(3)
      E=V0(4)**2/2.0D+00 - MU/R0(4)
      A=-1.0D+00/(2.0D+00*E)
      SMU=DSQRT(MU)
      TOL=1.0D-16
      X0=SMU * T/A
10    Z=X0**2/A
      SZ=DSQRT(Z)
      C=(1.d+00 - DCOS(SZ))/Z
      S=(SZ - DSIN(SZ))/SZ**3
      TN=DOT/SMU*X0**2*C + (1.d+00-R0(4)/A)*X0**3*S
     $      + R0(4)*X0
      DT=X0**2*C + DOT/SMU*X0*(1.d+00-Z*S) + R0(4)
     $      * (1.d+00-Z*C)
      XN=X0 + (T-TN)/DT
      IF ((XN-X0) .GT. TOL) THEN
         X0=XN
        GO TO 10
      ENDIF
      F=1.d+00 - XN**2/R0(4) * C
      G=XN**2 * DOT/MU * C + R0(4)*XN * (1.d+00-Z*S)/SMU
      DO 20 I=1,3
20      R(I)=F * R0(I) + G * V0(I)
      R(4)=DSQRT(R(1)**2+R(2)**2+R(3)**2)
      GD=1.d+00 - XN**2/R(4) * C
      FD=SMU/(R0(4)*R(4)) * XN * (Z*S-1.d+00)
      DO 30 I=1,3
30      V(I)=FD * R0(I) + GD * V0(I)
      RETURN
      END
       ************************************************************
```

Bibliography

1. Baker, Robert M.L., Jr. Astrodynamics. New York: Academic Press, 1967.

2. Barclay, Capt Richard C. A Computer Model for Evaluation of Launch Vehicle and Target Tracking Assignments for Direct Ascent, Deep Space ASAT Systems. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, December 1983.

3. Bate, Roger R., et al. Fundamentals of Astrodynamics. New York: Dover Publications, Inc., 1971.

4. Bowman, Bruce R. and Richard S. Hujsak. Low Thrust Spacecraft Acceleration Study for Survivability: Final Report, 14 February 1986. Contract F05604-85-C-0074. Applied Technology Associates, Inc., Delaware, Feb 1986.

5. Bowman, Robert M. "Arms Control in Space: Preserving Critical Strategic Space Systems Without Weapons in Space," Air University Review, 37(1): 58-72 (November-December 1985).

6. Brandt, MGen Robert A. "Military Use for Space," Air University Review, 37(1): 40-51 (November-December 1985).

7. Bryson, A.E. and Y.-C. Ho. Applied Optimal Control. Washington D.C.: Hemisphere Publishing Corporation, 1975.

8. Burk, Roger C. Minimum Impulse Orbital Evasive Maneuvers. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, December 1985.

9. Carter, Thomas E. "Fuel-Optimal Maneuvers of a Spacecraft Relative to a Point in Circular Orbit," Journal of Guidance, Control, and Dynamics, 9(1): 27-31 (January-February 1986).

10. Chadwick, C. "Computing Approximate Random Delta V Magnitude Probability Densities," AIAA/AAS Astrodynamics Conference. 1-10. AIAA Press, New York, 1984.

11. Cimbala, Stephen J. "The Strategic Defense Initiative Political Risks," Air University Review, 37(1): 24-35. (November-December 1985).

12. Finke, Robert C. Electric Propulsion and its Applications to Space Missions. Vol. 79: Progress in Astronautics and Aeronautics, New York: American Institute of Aeronautics and Astronautics, 1981.

13. Friedenstein, LTC Charles D. "The Uniqueness of Space Doctrine", Air University Review, 37(1): 13-22 (November-December 1985).

14. Greenwood, Donald T. Principles of Dynamics. New Jersey: Prentice-Hall, Inc., 1965.

15. Hintz, G.R. and C. Chadwick. "Design and Analysis Techniques for Trajectory Correction Maneuvers," AIAA/AAS Astrodynamics Conference. 1-11. AIAA Press, New York, 1984 (AIAA-84-2014).

16. Jensen, D.L. "Kinematics of Rendezvous Maneuvers," Journal of Guidance, Control, and Dynamics, 7(3): 307-314 (May-June 1984).

17. Jezewski, Donald J. "An Analytic Approach to Two-Fixed Impulse Transfers Between Keplerian Orbits," Journal of Guidance, Control, and Dynamics, 5(5): 458-464 (September-October 1982).

18. Kaplan, Marshall H. Modern Spacecraft Dynamics and Control. New York: John Wiley & Sons, Inc., 1976.

19. Kelley, Henry J. et al. "Pursuit/Evasion in Orbit," The Journal of Astronautical Sciences, 29(3): 277-288 (July-September 1981).

20. Koelle, Henry H. Handbook of Astronautical Engineering. New York: Magraw-Hill, Inc., 1961.

21. Lang, T.S. "Optimal Impulsive Maneuvers to Accomplish Small Plane Changes in Elliptical Orbit," Journal of Guidance, Control, and Dynamics, 2(4): 271-275 (July-August 1979).

22. Meirovitch, Leonard. Methods of Analytical Dynamics. New York: Magraw-Hill, Inc., 1970.

23. Smith, Eugene A. Minimum Fuel Continuous Low Thrust Orbital Transfer. MS Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, December 1974.

24. Stanford, R.H. "Constrained Maneuver Strategies for Project Galileo," AIAA/AAS Astrodynamic Conference. 1-7. AIAA Press, New York, 1984 (AIAA-84-2026).

25. Starr, J.L. and R.D. Sugar. "Some Computational Aspects of Minimum Fuel Continuous Low Thrust Orbit Transfer Problems." The Journal of the Astronautical Sciences, 19: 169-204 (November-December 1971).

26. Stuart, Dale G. "A Simple Targeting Technique for Two-Body Spacecraft Trajectories," Journal of Guidance, Control, and Dynamics, 9(1): 27-31 (January-February 1986).

27. Wagner, C.A. et al. "Orbit Targeting for Body Stabilized Synchronous Satellites with Attitude Constraints," AIAA 19th Aerospace Science Meeting. 1018-1027. AIAA Press, New York, 1981 (AIAA-81-0171).

28. Widhalm, Joseph W. Lecture materials from MC622 and MC624, Optimization Techniques and Trajectory Optimization. School of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, March, June 1986.

29. Wiesel, William E. Lecture materials from MC731, Modern Methods of Orbit Determination. School Of Engineering, Air Force Institute of Technology (AU), Wright Patterson AFB OH, March 1986.

30. Zarwonsky, Raymond M. "Intercept Probability of Satellites in Circular Orbits," Journal of Guidance, Control, and Dynamics, 8(1): 118-124 (January-February 1985).

VITA

Robert D. Preissinger was born 15 December, 1953, in Grand Prairie, Texas. He attended Joshua High School in Joshua, Texas, where he was valedictorian of his 1972 graduating class. An Air Force Academy appointment followed, where he received a Bachelor of Science degree in Engineering Sciences in June, 1976. Upon graduation, he was commissioned a second lieutenant and sent to Vance AFB for Undergraduate Pilot Training, where he was an outstanding graduate. After UPT, he served at Spangdahlem AB, West Germany, flying F-4C's, F-4D's, and F-4E's.

Upon returning to the United States, Capt Preissinger was assigned to Homestead AFB, Florida, as a F-4D instructor pilot. He instructed for three years, two years of which he was also a standardization evaluation flight instructor. Capt Preissinger was notified of acceptance to the Air Force Institute of Technology in the Astronautical Engineering program and entered the School of Engineering in May, 1985.

Permanent address: Route 3, Box 328
Joshua, Texas 76058

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

A179 413

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GA/AA/86D-12	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/ENY	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB, OH 45433	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
See Box 19.

12. PERSONAL AUTHOR(S)
Robert D. Preissinger, B.S., CPT, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
MS Thesis	FROM _____ TO _____	1986 December	76

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
22	03		Evasive Maneuvers, Thrust controls; Geosynchronous Satellite Orbits; Optimization; Continuous Thrust Programming. (Thesis)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Title: Optimal Continuous Thrust Orbital Evasive Maneuvers from Geosynchronous Orbit

Thesis Chairman: Joseph W. Widhalm, LtCol, USAF
Assistant Professor of Aerospace Engineering

Approved for public release: IAW AFR 190-1
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (AFIT)
Wright-Patterson AFB OH 45433

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Joseph W. Widhalm, LtCol, USAF	22b. TELEPHONE (Include Area Code) 513-255-4476 22c. OFFICE SYMBOL AFIT/ENY

DD Form 1473, JUN 86 Previous editions are obsolete. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED

A satellite under attack by another orbiting body relying on an explosive device to effect a kill has the problem of avoiding a volume of space in which its destruction is highly likely. To avoid this volume, the attackee could use continuous low thrust, such as that provided by electric propulsion or attitude control thrusters, if its warning time and orbital parameters were appropriate. A model is developed using optimal control theory and is solved numerically for the thrust direction using various magnitudes of thrust.

The model progresses from a one satellite solution (where the threat is a fixed sphere in space) to a two satellite solution in which the distance between the threat and the target satellite is maximized. A mininum time solution for the single satellite problem is included. The results are given for several values of thrust, several maneuvering times, the optimized time-of-maneuver, and for circular and non-circular end orbits.

Keywords: